**MODULE – 4(Advance PHP)**

**OOPS**

# What Is Object Oriented Programming?

-> Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

# What Are Properties Of Object Oriented Systems?

**-> 1) class**

* 1. **Object**
  2. **Encapsulation**
  3. **Abstraction**
  4. **Inheritance**
  5. **Polymorphism**

­­­­­­­­­­

# What Is Difference Between Class And

**Interface?**

# ->

## Class:

A class can be instantiated i.e., Objects of a class can be created.

Classes does not support multiple inheritance.

## Interface:

A Interface cannot be instantiated i.e., Objects cannot be created.

Interface supports multiple inheritance.

# What Is Overloading?

-> Overloading in PHP provides means to dynamically create properties and methods. These dynamic entities are processed via magic methods one can establish in a class for various action types.

# What Is T\_PAAMAYIM\_NEKUDOTAYIM (Scope Resolution Operator (::) with Example

-> The scope resolution operator also known as *Paamayim Nekudotayim* or more commonly known as the double colon is a token that allows access to static, constant, and overridden properties or methods of a class.

It is used to refer to blocks or codes in context to classes, objects, etc. An identifier is used with the scope resolution operator.

The most common example of the application of the scope resolution operator in PHP is to access the properties and methods of the class.

**Example:**

## <?php

**class democlass { const PI = 3.14;**

## }

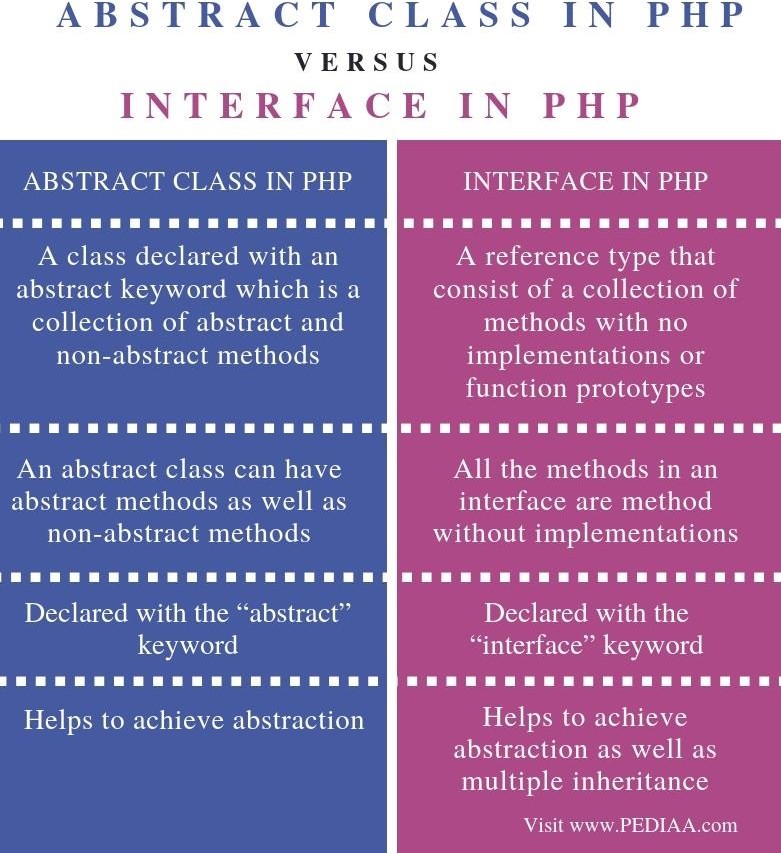
**echo democlass::PI;**

**?>**

# What are the differences between abstract

**classes and interfaces?**

# ->



* **Define Constructor and Destructor?**

# ->

**Constructors :**

Constructors are the blueprints for object creation providing values for member functions and member variables. Once the object is initialized, the constructor is automatically called.

**Destructors :**

Destructors are for destroying objects and automatically called at the end of execution. In this article, we are going to learn about object-oriented concepts of constructors and destructors.

Syntax:

* **construct():**

function construct()

{

// initialize the object and its properties by assigning

}

* **destruct():**

function destruct()

{

// destroying the object or clean up resources here

}

# How to Load Classes in PHP?

**->**

PHP load classes are used for declaring its object etc. in object oriented applications.

PHP parser loads it automatically, if it is registered With spl\_autoload\_register() function.

PHP parser gets the least chance to load class/interface before emitting an error.

Before using a class, you need to:

* **First, define the class in a file.**
* **Second, load it using the require, require\_once, include, or include\_once statement.**

Syntax:

spl\_autoload\_register(function ($class\_name)

{

include $class\_name . '.php';

});

**Example:**

**<?php**

**class Contact**

**{**

**private $email;**

**public function construct(string $email)**

**{**

**$this->email = $email;**

**}**

**public function getEmail()**

**{**

**return $this->email;**

**}**

**}**

# How to Call Parent Constructor?

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

If you create a construct() function, PHP will automatically call this function when you create an object from a class.

Notice that the construct function starts with two underscores ( )!

We see in the example below, that using a constructor saves us from calling the set\_name() method which reduces the amount of code:

**<?php class Fruit {**

**public $name; public $color;**

**function construct($name) {**

**$this->name = $name;**

**}**

**function get\_name() { return $this->name;**

**}**

**}**

**$apple = new Fruit("Apple"); echo $apple->get\_name();**

**?>**

# Are Parent Constructor Called Implicitly When Create An Object Of Class?

->Parent constructors are not called implicitly if the child class defines a constructor. In order to run a parent constructor, a call to parent:: construct() within the child constructor is required.

# What Happen, If Constructor Is Defined As Private Or Protected?

-> The constructor may be made private or protected to prevent it from being called externally.

If so, only a static method will be able to instantiate the class.

Because they are in the same class definition they have access to private methods, even if not of the same object instance.

Public, private and protected are called access modifiers.

Just like C++, PHP also have three access modifiers such as public, private and protected.

The visibility of a property, a method or a constant can be defined by prefixing the declaration with these keywords.

* **If the class member declared as public then it can be accessed everywhere.**
* **If the class members declared as protected then it can be accessed only within the class itself and by inheriting child classes.**
* **If the class members declared as private then it may only be accessed by the class that defines the member.**

# What are PHP Magic Methods/Functions? List them Write program for Static Keyword in PHP?

**->**

Magic methods are special methods which override PHP's default's action when certain actions are performed on an object. Caution.

All methods names starting with are reserved by PHP.

Therefore, it is not recommended to use such method names unless overriding PHP's behavior.

The following method names are considered magical:

construct(), destruct(), call(), callStatic(), get(),

set(), isset(), unset(), sleep(), wakeup(),

serialize(), unserialize(), toString(), invoke(),

set\_state(), clone(), and debugInfo().

## Example:

**<?php**

**class MyClass {**

**public static $str = "Hello World!";**

**public static function hello() { echo MyClass::$str;**

**}**

**}**

**echo MyClass::$str; echo "<br>";**

**echo MyClass::hello();**

**?>**

# Create multiple Traits and use it in to a single

**class ?**

# ->

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. (public, private, or protected).

# Write PHP Script of Object Iteration?

**->**

it is possible to iterate through list of all visible items of an object.

**Iteration can be performed using** foreach **loop as well as** iterator **interface.**

**There is also** IteratorAggregate **interface in PHP, that can be used for this purpose**

**Example:**

<?php

class myclass

{

private $var; protected $var1; public $x, $y, $z;

public function construct()

{

$this->var="private variable";

$this->var1=TRUE;

$this->x=100;

$this->y=200;

$this->z=300;

}

public function iterate()

{

foreach ($this as $key => $value)

{

print "$key => $value\n";