**1. What Is Object Oriented Programming?**

Object-oriented programming (OOP) is a programming paradigm that focuses on using objects, which are instances of classes, to represent real-world concepts and entities. OOP is based on the principles of abstraction, encapsulation, inheritance, and polymorphism.

Abstraction involves creating a simplified representation of a complex system by hiding the details of its implementation. Encapsulation is the practice of keeping data and behavior within an object to prevent outside access and modification. Inheritance allows a class to inherit properties and methods from another class, and polymorphism allows objects of different types to be treated as if they are the same type.

**2. What Are Properties Of Object Oriented Systems?**

The properties of object-oriented systems include:

Encapsulation: Objects keep their data and methods private, which prevents outside access and modification.

Inheritance: Classes can inherit properties and methods from other classes.

Polymorphism: Objects of different types can be treated as if they are the same type.

Abstraction: The complexity of a system is hidden behind a simplified representation.

Modularity: The system is made up of modules, which can be modified independently.

Dynamic Binding: The type of an object is determined at runtime, allowing for more flexible code.

**3. What Is Difference Between Class And Interface?**

A class is a blueprint for creating objects, while an interface is a contract that defines a set of methods that a class must implement.

A class can contain variables, methods, constructors, and other class members. When a class is instantiated, an object is created based on the class definition.

An interface, on the other hand, only contains method signatures and constants. It does not define the implementation of the methods, but rather specifies what methods a class must implement. A class can implement multiple interfaces, but can only inherit from one class.

**4. What Is Overloading?**

Overloading is a feature in object-oriented programming where two or more methods in a class have the same name, but different parameters.

When an overloaded method is called, the compiler determines which method to execute based on the number, type, and order of the parameters passed. This allows for more flexible and readable code, as methods can be named based on their functionality rather than their specific parameters.

**5. What Is T\_PAAMAYIM\_NEKUDOTAYIM, Scope Resolution Operator (::) with Example**

In PHP, T\_PAAMAYIM\_NEKUDOTAYIM denotes a syntax error while doing any task.

The reason for getting this error is only one. Paamayim Nekudotayim in Hebrew means double colon::

this error refers to the inappropriate use of the double colon operator (::). The error is typically caused by an attempt to call a static method that is, in fact, not static.

In PHP, the double colon :: is defined as Scope Resolution Operator. It used when when we want to access constants, properties and methods defined at class level.

class MyClass {

public static function myStaticMethod() {

return "Hello, World!";

}

}

echo MyClass::myStaticMethod();

**6. What are the differences between abstract classes and interfaces?**

Interface are similar to abstract classes. The difference between interfaces and abstract classes are:

- Interfaces cannot have properties, while abstract classes can

- All interface methods must be public, while abstract class methods is public or protected

- All methods in an interface are abstract, so they cannot be implemented in code and the abstract keyword is not necessary

- Classes can implement an interface while inheriting from another class at the same time

**7. Define Constructor and Destructor?**

Constructors: Constructors are called when an object is created from a class.A constructor is a special member function that is called automatically when an object is created. Its primary purpose is to initialize the object's data members and allocate any necessary resources for the object to function correctly.

Destructors: Destructors are called when an object destructs. Usually, it is when the script ends.A destructor is a special member function that is called automatically when an object is destroyed or goes out of scope. Its primary purpose is to free up any resources allocated to the object during its lifetime, such as memory or file handles.

**8. 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.

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

include $class\_name . '.php';

});

**9. How to Call Parent Constructor?**

In PHP, you can call the constructor of the parent class using the parent::\_\_construct() method within the constructor of the child class. This allows you to inherit and use the constructor of the parent class when creating objects of the child class.

<?php

class grandpa{

public function \_\_construct(){

echo "I am in Tutorials Point"."

";

}

}

class papa extends grandpa{

public function \_\_construct(){

parent::\_\_construct();

echo "I am not in Tutorials Point";

}

}

$obj = new papa();

?>

**10. Are Parent Constructor Called Implicitly When Create An Object Of Class?**

No, the constructor of the parent class is not called implicitly when an object of a child class is created in PHP.

When you create an object of a child class, PHP only calls the constructor of that child class. If you want to also call the constructor of the parent class, you need to explicitly call it using the parent::\_\_construct() method within the constructor of the child class, as shown in my previous answer.

It's important to note that if the parent constructor has required parameters, you need to pass those parameters when calling parent::\_\_construct(). If you don't pass the required parameters, PHP will raise an error.

**11. What Happen, If Constructor Is Defined As Private Or Protected?**

If a constructor is defined as private or protected in PHP, it cannot be called from outside the class, including from any child classes.

A private constructor can only be called from within the same class where it is defined. This means that you cannot create an object of the class from outside the class, and any attempts to do so will result in a fatal error.

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

PHP magic methods, also known as magic functions, are special methods that start with a double underscore ("\_\_") in their name. These methods are automatically called by PHP in response to certain events, such as object creation, property access, and method calls.

Here is a list of some of the most commonly used magic methods in PHP:

\_\_construct(): This magic method is called automatically when an object of a class is created, and it is used to initialize the object's properties.

\_\_destruct(): This magic method is called automatically when an object of a class is destroyed, and it is used to perform any necessary cleanup tasks.

\_\_get(): This magic method is called automatically when an undefined or inaccessible property is accessed, and it is used to dynamically get the value of the property.

\_\_set(): This magic method is called automatically when an undefined or inaccessible property is set, and it is used to dynamically set the value of the property.

\_\_isset(): This magic method is called automatically when an undefined or inaccessible property is checked for existence using the isset() function, and it is used to dynamically determine if the property exists.

\_\_unset(): This magic method is called automatically when an undefined or inaccessible property is unset using the unset() function, and it is used to dynamically unset the property.

\_\_call(): This magic method is called automatically when an undefined or inaccessible method is called, and it is used to dynamically handle the method call.

\_\_toString(): This magic method is called automatically when an object of a class is used in a string context, such as when it is concatenated with a string, and it is used to return a string representation of the object.

class MyClass {

public static $count = 0;

public function \_\_construct() {

self::$count++;

}

public static function getCount() {

return self::$count;

}

}

$obj1 = new MyClass();

$obj2 = new MyClass();

$obj3 = new MyClass();

echo MyClass::getCount();

**13. Create multiple Traits and use it in to a single class?**

trait TraitA {

public function methodA() {

echo "Method A from Trait A\n";

}

}

trait TraitB {

public function methodB() {

echo "Method B from Trait B\n";

}

}

class MyClass {

use TraitA, TraitB;

public function methodC() {

echo "Method C from MyClass\n";

}

}

$obj = new MyClass();

$obj->methodA();

$obj->methodB();

$obj->methodC();

**14. Write PHP Script of Object Iteration?**

From PHP 5 onwards, 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

<?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

";

}

}

}

$obj = new myclass();

foreach($obj as $key => $value) {

print "$key => $value

";

}

echo "

**15. Use of The $this keyword**

In PHP, the $this keyword is a reference to the current object instance inside a class. It is used to access properties and methods of the object, and can be used to pass the object as a parameter to other methods.

class Person {

public $name;

public $age;

public function \_\_construct($name, $age) {

$this->name = $name;

$this->age = $age;

}

public function greet() {

echo "Hello, my name is {$this->name} and I am {$this->age} years old.";

}

}

$person = new Person("John Doe", 30);

$person->greet();

**16. Consider the exercise11and add a edit link near delete link e.g. Clicking up on edit button a particular row should be open in editing mode e.g. on the Particular row there should be filled text box with data and on the option column there should be a confirm button clicking upon it arrow should be updated.**

**17. Create Hotel Room Booking System User can book room by 3 ways Full day Half day Custom.**

**18. If user select for the full day than user only have selection for the checking checkout date**

**19. If user select Half day than user have option of date and slot option (like user want to book room for first half – Morning (8AM to 6PM) if user select for second halfit‟s for evening (7PM to Morning 7AM)). Do proper validation like if user can book only available slot. (have touse jQuery -> Ajax, validation, Json passing).**