1. **What is Object-Oriented Programming?**

**ANS:--**

Object-Oriented Programming (OOP) is a programming paradigm based on the concept of "objects." Objects are instances of classes and represent real-world entities with attributes (properties) and behaviors (methods). OOP focuses on encapsulating data and methods inside objects, promoting modularity, code reuse, and abstraction.

**2. What Are the Properties of Object-Oriented Systems?**

ANS:--

* **Encapsulation**: Bundling of data (attributes) and methods (functions) inside a single unit, i.e., an object.
* **Inheritance**: The ability of one class (child) to inherit properties and behaviors (methods) from another class (parent).
* **Polymorphism**: The ability to take many forms, allowing methods to do different things based on the object that calls them.
* **Abstraction**: Hiding complex implementation details and exposing only the necessary parts.

**3. What Is the Difference Between Class and Interface?**

ANS:--

* **Class**: A blueprint for creating objects. It can contain both concrete methods (with implementation) and data members.
* **Interface**: A contract that defines method signatures but has no implementation. Classes that implement interfaces must provide an implementation for all the methods declared in the interface.

**4. What Is Overloading?**

ANS:-- 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. In other terms creating. Both method calls and member accesses can be overloaded via the call, get and set methods. These methods will only be triggered when your object or inherited object doesn't contain the member or method you're trying to access. All overloading methods must not be defined as static. All overloading methods must be defined as public.

**5. What Is T\_PAAMAYIM\_NEKUDOTAYIM (Scope Resolution Operator ::) with Example**  
The T\_PAAMAYIM\_NEKUDOTAYIM error occurs when there’s a misuse of the scope resolution operator (::). The :: operator is used to access static, constant, or overridden properties or methods of a class.

Example:

class MyClass {

public static $var = "Hello";

public static function sayHello() {

echo "Hello World!";

}

}

// Accessing static property and method

echo MyClass::$var;

MyClass::sayHello();

**7. Define Constructor and Destructor?**

**ANS:--**

* **Constructor**: A special method that is called automatically when an object is created. It is usually used to initialize object properties.
* **Destructor**: A method that is called automatically when an object is destroyed (usually at the end of a script execution or when explicitly called).

**8. How to Load Classes in PHP?**  
Classes can be loaded using the require, include, require\_once, or include\_once statements. For automatic class loading, you can use **autoloading** by implementing spl\_autoload\_register().

**9. How to Call Parent Constructor?**  
You can call the parent constructor inside a child class using parent::\_\_construct().

**10. Are Parent Constructors Called Implicitly When Creating an Object of a Child Class?**  
No, parent constructors are not called implicitly. You need to explicitly call the parent constructor using parent::\_\_construct().

**11. What Happens If a Constructor Is Defined as Private or Protected?**

* **Private**: The constructor can only be called within the same class, preventing object instantiation from outside the class.
* **Protected**: The constructor can be called within the class and its subclasses, but not from outside.

**12. What Are PHP Magic Methods/Functions? List Them.** Magic methods in PHP are special methods that start with double underscores (\_\_). They are used to perform various operations like overloading, serialization, and more. Some common magic methods include:

* \_\_construct(): Constructor
* \_\_destruct(): Destructor
* \_\_call(): Called when an inaccessible method is called
* \_\_get(), \_\_set(): For accessing or setting inaccessible properties
* \_\_toString(): Converts an object to a string

**13. Write a Program for Static Keyword in PHP?**

Code:

class MyClass {

public static $counter = 0;

public function \_\_construct() {

self::$counter++;

}

public static function getCount() {

return self::$counter;

}

}

$obj1 = new MyClass();

$obj2 = new MyClass();

echo MyClass::getCount(); // Output: 2

**14. Create Multiple Traits and Use Them in a Single Class?**

Code:--

trait TraitOne {

public function methodOne() {

echo "Trait One method\n";

}

}

trait TraitTwo {

public function methodTwo() {

echo "Trait Two method\n";

}

}

class MyClass {

use TraitOne, TraitTwo;

}

$obj = new MyClass();

$obj->methodOne(); // Output: Trait One method

$obj->methodTwo(); // Output: Trait Two method

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

Code:--

class MyClass {

public $prop1 = "Property 1";

public $prop2 = "Property 2";

public function iterateProperties() {

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

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

}

}

}

$obj = new MyClass();

$obj->iterateProperties();

**16. Use of the $this Keyword?**

**ANS:--** The $this keyword refers to the current object instance in a class. It's used to access the object's properties and methods.

**17. Consider the Exercise... Editing Mode**

ANS:--This refers to adding an edit functionality in a table with a form to update the data, typically done with jQuery, Ajax, and PHP backend handling.