**1) What is OOPS?**

OOPS is abbreviated as Object Oriented Programming system in which programs are considered as a **collection of objects**.

**2) What is a class?**

A class is simply a **representation** of a **type of object**. It is the **blueprint**/plan/template that describes the details of an **object**.

OR

A class is a template for objects, and an object is an instance of class.

**3) What is an Object?**

An object is an **instance** of a **class**. It has its own state, behavior, and identity.

**4) What is Inheritance?**

Inheritance is a concept where **one class shares** the structure and behavior defined in **another class**. If Inheritance applied to one class is called Single Inheritance, and if it depends on multiple classes, then it is called multiple Inheritance.

OR

Inheritance in OOP = When a class derives from another class.

**5) Explain the term constructor**

A constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules for constructor are:

* Constructor Name should be the same as a class name.
* A constructor must have no return type.

OR

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

**6) What are the access modifiers?**

Access modifiers determine the scope of the method or variables that can be accessed from other various objects or classes. There are three types of access modifiers, and they are as follows:

* Private
* Protected
* Public

public :- use from anywherere  eg.outside class ,inside class ,derived class also

protected :- use in class iteself and derived class only

private :-  only class itself

**7) What is Method/Function Overriding?**

In function overriding, both parent and child classes should have same function name with and number of arguments.

**8) What is Properties Overriding?**

In **Properties** overriding, both parent and child classes should have same **Properties** name with and arguments.

**9) What is Abstract Class?**

An abstract class is a class that contains at least one abstract method. An abstract method is a method that is declared, but not implemented in the code.

**10) What is Interfaces?**

Interfaces allow you to specify what methods a class should implement.

Interfaces make it easy to use a variety of different classes in the same way. When one or more classes use the same interface, it is referred to as "polymorphism".

**10) What is Static Methods?**

Static methods can be called directly - without creating an instance of the class first.

Static methods are declared with the static keyword:

To access a static method use the class name, double colon (::), and the method name:

**10) What is Static properties?**

Static properties can be called directly - without creating an instance of a class.

Static properties are declared with the static keyword:

To access a static property use the class name, double colon (::), and the property name.

**11) What is 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 (public, private, or protected).

**11) What is Type Hinting?**

* In simple word, type hinting means providing hints to function to only accept the given data type.
* In technical word we can say that Type Hinting is method by which we can force function to accept the desired data type.
* In PHP, we can use type hinting for Object, Array and callable data type.

**11) What is Namespaces?**

1. They allow for better organization by grouping classes that work together to perform a task
2. They allow the same name to be used for more than one class

**11) What is Method Chaining?**

This method is called Method Chaining in PHP’s terminology. Each method in class in Method Chaining, that is, the method of the class returns the object of that class. For Method Chaining, instead of writing value return in class, we have to write **return $this;**.

**12) What is Method Destructor?**

A destructor is called when the object is destructed or the script is stopped or exited.

If you create a \_\_destruct() function, PHP will automatically call this function at the end of the script.

Notice that the destruct function starts with two underscores (\_\_)!

The example below has a \_\_construct() function that is automatically called when you create an object from a class, and a \_\_destruct() function that is automatically called at the end of the script:

**13) What is Magic Method Autoload?**

**Autoloading** is the process of automatically loading **PHP** classes without explicitly loading them with the require() , require\_once() , include() , or include\_once() functions.

**14) What is Magic Method \_\_get ()?**

\_\_get() is utilized for reading data from inaccessible (protected or private) or non-existing properties.

.

**15) What is Magic Method \_\_set ()?**

Usually in php there are certain circumstances when you need to access a property directly by its name but because the property you want to access declared as protected or private it will cause error so to access it you have to implement  \_\_get() and \_\_set()..

**16) What is Magic Method \_\_call ()?**

The \_\_call() method will be called when you try to access a method that is not publicly accessible on the object.

**17) What is Magic Method \_\_callStatic ()?**

Like the **\_\_call()** the **\_\_callStatic()** magic method used when you attempt to access a non public method, but the difference between them is that when you call the method in static scope for example Class::method().

**18) What is Magic Method \_\_isset ()?**

You already encountered the isset() function in a lot of scripts when working with arrays to check if specific key is exist. You can also use this function on objects to see if a publicly accessible property has been set.

**19) What is Magic Method \_\_unset ()?**

On the other hand, the \_\_unset() is a method called when you call the unset() method on inaccessible or non-existent object properties.

**20) What is Magic Method \_\_toString ()?**

The \_\_toString() magic method allows you to define what you would like to display when an object of the class is treated like a string. If you use echo or print on your object, and you haven’t defined the \_\_toString() method, it’ll give an error.

**21) What is Magic Method \_\_toString ()?**

The \_\_toString() magic method allows you to define what you would like to display when an object of the class is treated like a string. If you use echo or print on your object, and you haven’t defined the \_\_toString() method, it’ll give an error.

**22) What is Magic Method \_\_sleep ()?**

The \_\_sleep() magic method is different compared to the methods that we’ve discussed so far. It’s called when you call the serialize() function on the object. In the case of a very large object, you only want to save selected properties during serialization and clean up the object.The \_\_sleep() method must return an array with the names of all properties of the object that should be serialized.

**23) What is Magic Method \_\_wakeup ()?**

On the other hand, the use of the \_\_wakeup() magic method is to re-establish any connections and start up tasks when the unserialize() function is called on the object.

**24) What is Magic Method \_\_clone ()?**

If you want to duplicate an existing object, you could use the clone keyword to do that. But after cloning, if you want to modify properties of the cloned object, you can define the \_\_clone() magic method in your class.

**24) What is Magic Method** \_\_invoke **()?**  
The \_\_invoke() magic method is a special method which is called when you try to call an object as if it were a function. Firstly, let’s see how it works, and then we’ll see the purpose of this magic method.

**25) What is** Magic Constants**?**  
1.[\_\_LINE\_\_](https://www.javatpoint.com/php-magic-constants#LINE)

1. [\_\_FILE\_\_](https://www.javatpoint.com/php-magic-constants#FILE)
2. [\_\_DIR\_\_](https://www.javatpoint.com/php-magic-constants#DIR)
3. [\_\_FUNCTION\_\_](https://www.javatpoint.com/php-magic-constants#FUNCTION)
4. [\_\_CLASS\_\_](https://www.javatpoint.com/php-magic-constants#CLASS)
5. [\_\_TRAIT\_\_](https://www.javatpoint.com/php-magic-constants#TRAIT)
6. [\_\_METHOD\_\_](https://www.javatpoint.com/php-magic-constants#METHOD)

8.[\_\_NAMESPACE\_\_](https://www.javatpoint.com/php-magic-constants#NAMESPACE)