**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).