**OOP**

● What is object-oriented programming in general terms?

Object-oriented programming **combines a group of data attributes with functions or methods into a unit called an "object."** Typically, OOP languages are class-based, which means that a class defines the data attributes and functions as a blueprint for creating objects, which are instances of the class.

● What is a class?

A class is a user-defined type that describes what a certain type of object will look like. A class description consists of a declaration and a definition. Usually these pieces are split into separate files. An object is a single instance of a class. You can create many objects from the same class type

● What is an object?

In PHP, Object is **a compound data type (along with arrays)**. Values of more than one types can be stored together in a single variable. Object is an instance of either a built-in or user defined class. In addition to properties, class defines functionality associated with data.

● What is an instance?

An instance is **an object that has been created from an existing class**. Creating an object from an existing class is called instantiating the object. To create an object out of a class, the new keyword must be used. Classes should be defined prior to instantiation

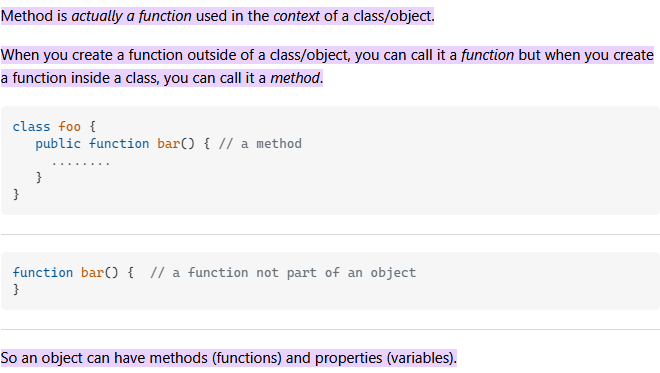
● What is a property?

Classes can have variables within it. Those variables are called properties. **A property is a normal PHP variable which is in any data type (integer, string, array, object, etc)**. In classes, before declaring a variable, we should add the visibility keyword to define where the variable is available

● What is a method?

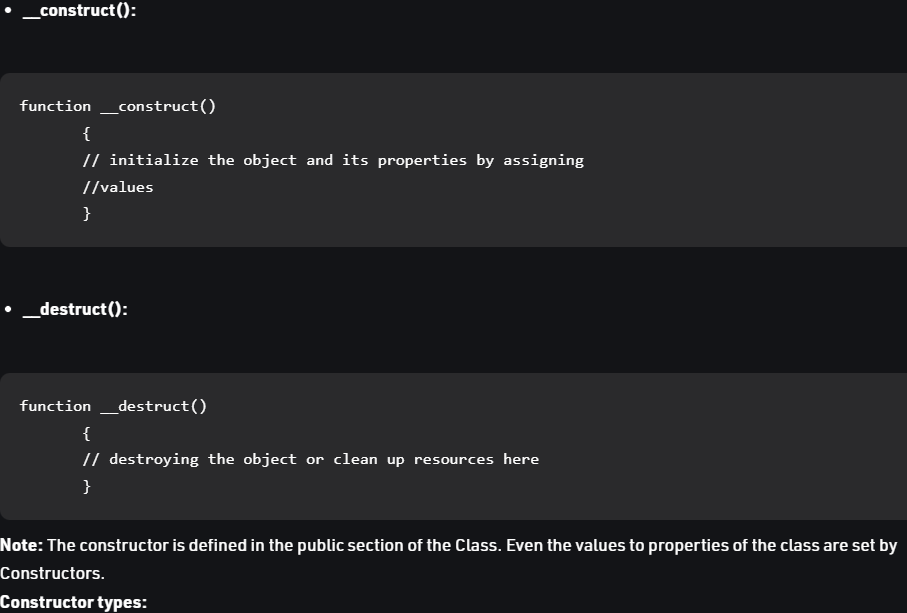
In Object Oriented Programming in PHP, methods are **functions inside classes**. Their declaration and behavior are almost similar to normal functions, except their special uses inside the class.

● What is the difference between a function and a method?



● What is a constructor?

Constructors are special member functions for initial settings of newly created object instances from a class, which is the key part of the object-oriented concept in **PHP5**.  
Constructors are the very basic building blocks that define the future object and its nature. You can say that the 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 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.   
Both are special member functions of any class with different concepts but the same name except destructors are preceded by a **~ Tilda operator**.

**Syntax:** 

● What is the difference between a class, an object and an instance?

A class is what you use to **define** the properties, methods and behavior of objects. Objects are the **things you create** out of a class. Think of a class as a blueprint, and an object as the actual building you build by following the blueprint (class). (Yes, I know the blueprint/building analogy has been done to death.

● What do we understand about the concept of encapsulation?

So the OOPs concept of Encapsulation in PHP means, **enclosing the internal details of the object to protect from external sources**. It describes, combining the class, data variables and member functions that work on data together within a single unit to form an object.

● What do we understand about the concept of abstraction?

PHP has abstract classes and methods. **Classes defined as abstract cannot be instantiated, and any class that contains at least one abstract method must also be abstract**. Methods defined as abstract simply declare the method's signature; they cannot define the implementation.

● What do we understand about the concept of inheritance?

PHP - What is Inheritance? Inheritance in OOP = **When a class derives from another class**. The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods.

● What do we understand about the concept of polymorphism?

Polymorphism **allows objects of different classes to respond differently based on the same message**. To implement polymorphism in PHP, you can use either abstract classes or interfaces. Polymorphism helps you create a generic framework that takes the different object types that share the same interface.

● What do we understand about the concept of Overload?

Overloading is also such a concept of OOPs in PHP. Function overloading or method overloading is a feature that allows creating several methods with the same name, which differ from each other in the type of input parameters. It is simply defined as **the ability of one function to perform different tasks**.

● What do we understand about the concept of Override?

Overriding is **an Object-Oriented Programming concept that is similar to a concept like Class, Object, Encapsulation, Polymorphism, Overloading etc in PHP**. Overriding of functions and classes are done when a method in the derived class is created which is the same as that of the method in the base class or parent class.

● What differences exist between the concept of Overload and Override?

Function overloading and overriding is the OOPs feature in PHP. In function overloading, more than one function can have same method signature but different number of arguments. But in case of function overriding, more than one functions will have same method signature and number of arguments.

● What is a static class?

Introduction: A static class in PHP is **a type of class which is instantiated only once in a program**. It must contain a static member (variable) or a static member function (method) or both. The variables and methods are accessed without the creation of an object, using the scope resolution operator(::).

● Look for 3 advantages over object-oriented programming compared to other

programming paradigms

OOP is **faster and easier to execute**. OOP provides a clear structure for the programs. OOP helps to keep the PHP code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug. OOP makes it possible to create full reusable applications with less code and shorter development time.

● Look for disadvantages of this paradigm.

Some of the disadvantages of object-oriented programming include: **Steep learning curve**: The thought process involved in object-oriented programming may not be natural for some people, and it can take time to get used to it. It is complex to create programs based on interaction of objects.