**Advance php**

**OOPs Concepts**

**Define Object-Oriented Programming (OOP) and its four main principles: Encapsulation, Inheritance, Polymorphism, and Abstraction.**

Object-Oriented Programming (OOP) in PHP is a programming paradigm that structures code around objects, which are instances of classes. Each object bundles **data** (attributes/properties) and **behavior** (methods/functions).

The goal of OOP is to make programs more **modular, reusable, and easier to maintain**.

**Four Main Principles of OOP**

1. **Encapsulation**

**Definition** :- The practice of **hiding internal details** of an object and only exposing necessary functionality.

**How it works**:- Data (variables) and methods (functions) are wrapped together in a class. Access is often controlled using access modifiers (public, private, protected).

**Example**: A BankAccount class hides the balance field (private) and provides methods like deposit() and withdraw() to safely modify it.

1. **Inheritance**

**Definition**:- The mechanism by which one class (child/subclass) can **acquire the properties and behaviors** of another class (parent/superclass).

**How it works**:- Promotes **code reusability** and a hierarchical classification.

**Example**:- A Car class can inherit from a more general Vehicle class, gaining attributes like speed and methods like start(), while also adding its own unique behavior.

1. Polymorphism

**Definition**:- The ability of a single function, method, or operator to **behave differently based on context**.

**Types**:

* **Compile-time (method overloading):** Same method name but different parameter lists.
* **Runtime (method overriding):** Subclass provides a specific implementation of a method already defined in the parent class.

Example:

**Overloading**: add(int a, int b) and add(double x, double y)

**Overriding**: A Dog class overrides the makeSound() method of the Animal class to bark instead of making a generic sound.

1. Abstraction

**Definition:** The process of hiding complex implementation details and showing only the essential features of an object.

**How it works**: Achieved using abstract classes or interfaces. The focus is on **what an object does, not how it does it.**

**Example:** A Shape interface may define methods like draw() and area(). Concrete classes like Circle and Rectangle implement these methods differently but expose the same abstract behavior.

**Class**

**Explain the structure of a class in PHP, including properties and methods.**

A class is defined by using the [class](https://www.w3schools.com/php/keyword_class.asp) keyword, followed by the name of the class and a pair of curly braces ({}). All its properties and methods go inside the braces:

Syntax:

<?php

Class class\_name{

// code here

}

?>

Below we declare a class named fruit consisting of two properties ($name and $color) and two methods set\_name() and get\_name() for setting and getting the $name property:

<?php

Class fruit{

// properties

Public $name;

Public $color;

//methods

Function set\_name($name)

{

$this->name=$name;

}

Function get\_name(){

Return $this->name;

}

}

?>

**Note:** In a class, variables are called properties and functions are called methods!

**Properties (Variables):**

Properties are variables that belong to a class and define the attributes or state of an object created from that class. They are declared within the class using an access modifier (public, protected, or private) followed by the $ symbol and the property name.

<?php

class Car {

public $color; // Public property, accessible from anywhere

protected $model; // Protected property, accessible within the class and its child classes

private $year; // Private property, accessible only within the class

}

?>

Methods are functions defined within a class that define the behavior or actions an object can perform. They are declared using an access modifier (public, protected, or private) followed by the function keyword, the method name, and parentheses () for arguments.

<?php

class Car {

public $color;

public function setColor($newColor) { // Public method

$this->color = $newColor;

}

protected function getModel() { // Protected method

return "This is a car model.";

}

private function calculateAge() { // Private method

return 2025 - $this->year; // Assuming $this->year is a private property

}

}

?>

* **Access Modifiers:**

public, protected, and private control the visibility and accessibility of properties and methods.

* + public: Accessible from anywhere.
  + protected: Accessible within the class itself and by its child classes.
  + private: Accessible only within the defining class.
* $this Keyword:

Within a class method, $this refers to the current object instance, allowing access to its properties and other methods.

Object

**What is an object in OOP? Discuss how objects are instantiated from classes in PHP.**

Classes are nothing without objects! We can create multiple objects from a class. Each object has all the properties and methods defined in the class, but they will have different property values.

Objects of a class are created using the [new](https://www.w3schools.com/php/keyword_new.asp) keyword.

<?php

class Car {

public $model;

public $year;

// Constructor to initialize properties

public function \_\_construct($model, $year) {

$this->model = $model;

$this->year = $year;

}

// Method to display details

public function getDetails() {

return "Model: " . $this->model . ", Year: " . $this->year;

}

}

// Object instantiation

$car1 = new Car("Toyota Corolla", 2022); // first object

$car2 = new Car("Honda Civic", 2023); // second object

// Access methods through objects

echo $car1->getDetails(); // Output: Model: Toyota Corolla, Year: 2022

echo "<br>";

echo $car2->getDetails(); // Output: Model: Honda Civic, Year: 2023

?>

$car1 & $car2 is the instances of a class car.

Extends

**Explain the concept of inheritance in OOP and how it is implemented in PHP.**

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.

An inherited class is defined by using the [extends](https://www.w3schools.com/php/keyword_extends.asp) keyword.