# **Chapter: Introduction to PHP OOP — Class, Object, Access Specifiers, and Member Variables**

## **1. What is Object-Oriented Programming (OOP)?**

**Object-Oriented Programming (OOP)** is a style of programming where we organize code into **classes** and **objects**.  
 It helps in **reusability**, **scalability**, and **organization** of code.

## **2. Defining a Class**

A **class** is a blueprint for creating objects.  
 It defines **properties (variables)** and **methods (functions)** that describe the behavior of the object.

**Syntax:**

class ClassName {

// Properties (Member Variables)

// Methods

}

**Example:**

<?php

class Car {

// Member variables

public $brand;

public $color;

// Method

public function displayInfo() {

echo "Brand: $this->brand, Color: $this->color";

}

}

?>

## **3. Creating an Object**

An **object** is an **instance** of a class.  
 We use the new keyword to create an object.

**Syntax:**

$objectName = new ClassName();

**Example:**

<?php

// Creating an object

$myCar = new Car();

// Assigning values to member variables

$myCar->brand = "Toyota";

$myCar->color = "Red";

// Calling a method

$myCar->displayInfo();

?>

**Output:**

Brand: Toyota, Color: Red

## **4. Access Specifiers (Modifiers)**

**Access Specifiers** (also called **Access Modifiers**) define the visibility of class properties and methods.

| **Specifier** | **Meaning** |
| --- | --- |
| public | Accessible from anywhere. |
| protected | Accessible only within the class and its subclasses. |
| private | Accessible only within the class itself. |

## **5. Member Variables**

**Member Variables** are variables declared inside a class to store data.

### 

### 

### 

### 

### 

### 

### **Example showing public, protected, and private:**

<?php

class Person {

public $name; // Public property

protected $age; // Protected property

private $salary; // Private property

// Public method to set age and salary

public function setDetails($age, $salary) {

$this->age = $age;

$this->salary = $salary;

}

// Public method to get salary

public function getSalary() {

return $this->salary;

}

}

?>

## **6. Accessing Properties and Methods**

**Public:** Can be accessed directly.

<?php

$person = new Person();

$person->name = "John"; // OK

echo $person->name; // OK

?>

**Protected and Private:** Cannot be accessed directly from outside the class.

<?php

// This will cause an error

// $person->age = 25; // Error

// echo $person->salary; // Error

?>

**Correct way (through public methods):**

<?php

$person->setDetails(25, 50000); // Set age and salary

echo $person->getSalary(); // Get salary

?>

**Output:**

50000

## **7. Summary Table**

| **Term** | **Description** | **Example** |
| --- | --- | --- |
| Class | Blueprint of an object | class Car {} |
| Object | Instance of a class | $myCar = new Car(); |
| Access Specifier | Controls access (public/protected/private) | public $name; |
| Member Variables | Variables inside a class | $brand, $color |

# **Conclusion**

* **Class** defines the structure.
* **Object** is an actual entity based on a class.
* **Access Specifiers** control who can access variables or methods.
* **Member Variables** store data about the object.

By mastering these basic building blocks, you are ready to dive deeper into concepts like **constructors**, **inheritance**, **polymorphism**, and **interfaces** in PHP OOP.

# **Short Practice Exercises**

### **Exercise 1: Create a Basic Class and Object**

* Create a class named Animal with two public properties: type and sound.
* Create an object from the Animal class.
* Set the type to "Dog" and sound to "Bark".
* Display the values.

### **Exercise 2: Use Access Specifiers**

* Create a class named Employee.
* Add:  
  + a **public** property $name
  + a **protected** property $position
  + a **private** property $salary
* Create a public method setDetails() to set all three properties.
* Create a public method getSalary() to return the salary.
* Create an object and set details, then print the employee name and salary.

### **Exercise 3: Create Multiple Objects**

* Create a class Book with public properties: title and author.
* Create two objects: one for "Harry Potter" by "J.K. Rowling" and another for "The Alchemist" by "Paulo Coelho".
* Display details for both books using a method displayBook().

### 

### 

### **Exercise 4: Access Control Testing**

* Create a class BankAccount with:  
  + a **public** property $accountHolder
  + a **private** property $balance
* Create methods:  
  + deposit($amount)
  + getBalance()
* Create an object, deposit some amount, and display the balance.

# **Quick Hints for Solving**

* Use public, protected, private properly.
* Use $this->propertyName inside methods.
* Create objects using new ClassName().
* Call methods using $object->methodName().

# **Sample Mini Quiz**

Answer in 1 line:

**Q1:** What keyword is used to create an object?  
 **Q2:** Which access specifier allows properties to be accessed anywhere?  
 **Q3:** Can a private variable be accessed outside the class directly? (Yes/No)