ELEVO LABS
LEARN. INNOVATE. ELEVATE.

Course Overview

Course Details

Course Name: PHP OOP: Object-Oriented Programming for Modern PHP Development

Instructor: Michael Lee

Created On: 16 June, 2025

Updated On: 16 June, 2025

Price: 2500 INR

Duration: 2 Weeks

Modules: 10

Language: English, Hindi

Level: Intermediate, Expert

Certifications: Yes

Course Description

Elevate your PHP development skills by mastering Object-Oriented Programming (OOP). This comprehensive course is designed for developers who have a solid grasp of PHP fundamentals and are ready to transition to writing more structured, maintainable, and scalable code using OOP principles.

You will begin by understanding the core concepts of OOP: classes, objects, properties, and methods, along with the crucial `$this` keyword, constructors, and destructors. We then delve deep into the four pillars of OOP: Encapsulation, Inheritance, Polymorphism, and Abstraction, learning how to apply them effectively using access modifiers, abstract classes, and

interfaces.

The curriculum extends to advanced OOP features unique to PHP, such as traits for code reuse, static members, class constants, and powerful magic methods that allow you to customize object behavior. You'll learn how to organize your code with namespaces and implement robust exception handling in an OOP context. Finally, we'll introduce SOLID Principles, Dependency Injection, Composition, and common OOP design patterns, providing you with architectural blueprints for building robust applications.

What You'll Learn:

- Core OOP concepts: Classes, Objects, Properties, Methods, Constructors, and Destructors.
- Encapsulation: Implementing `public`, `private`, `protected` access modifiers, and using getters/setters.
- Inheritance: Using `extends`, `parent` keyword, and Method Overriding; understanding `final` classes/methods.
- Polymorphism: Understanding the concept, and achieving it with Abstract Classes and Interfaces (including multiple interface implementation).
- Abstraction: Hiding complexity with abstract classes and interfaces.
- Static Properties/Methods and Class Constants (`self` vs. `static`).
- Traits for flexible code reuse.
- Namespaces for code organization and Autoloading (`spl_autoload_register()`).
- Magic Methods: `__get`, `__set`, `__call`, `__callStatic`, `__toString`, `__invoke`, `__isset`, `__unset`, `__clone`, `__debugInfo`, `__sleep`, `__wakeup`.
- Advanced Exception Handling in an OOP context.
- Dependency Injection (DI): Principles and practical implementation.
- Design Patterns: Common OOP solutions (Singleton, Factory, Strategy, Observer, etc.).
- SOLID Principles: Best practices for writing maintainable OOP code.
- Composition vs. Inheritance: Understanding their differences and when to use each.
- Type Hinting & Return Types: Enforcing data integrity.
- Late Static Binding: Understanding dynamic static method calls.
- Reflection API: Inspecting code at runtime.
- Object Cloning: Deep vs. Shallow copies.
- Writing clean, modular, and scalable PHP code following modern OOP practices.

Who Is This Course For?

This course is for PHP developers who have a solid understanding of PHP fundamentals (e.g., from 'PHP & MySQL Fundamentals' course or equivalent procedural PHP experience) and are looking to transition to or deepen their knowledge of Object-Oriented Programming. It's essential for anyone aiming to work with modern PHP frameworks like Laravel or Symfony, or build larger, more maintainable applications.

Prerequisites:

- **Solid PHP Fundamentals:** A strong understanding of PHP syntax, variables, data types, control structures, functions, and arrays is essential.
- **Basic Understanding of Web Concepts:** Familiarity with HTTP, forms, and client-server interaction (as covered in a fundamentals course).
- **A Local PHP Development Environment:** Such as XAMPP, MAMP, or Laragon.
- **A Computer/Laptop:** For coding practice.
- **Willingness to Learn:** Enthusiasm for mastering advanced programming concepts.

Includes practical coding challenges for each principle, mini-projects applying OOP concepts, and a final capstone project. Certificate of completion available. Available in English and Hindi.

Course Curriculum Details

Module 1: Module 1: Introduction to OOP in PHP

7 Lessons

1. 1. Why Object-Oriented Programming? (Procedural vs. OOP Benefits) **20min**
2. 2. Core Concepts: Classes, Objects, Properties, and Methods **30min**
3. 3. Defining Classes and Creating Objects **25min**
4. 4. The `$this` Keyword: Referring to the Current Object **20min**
5. 5. Constructors (`__construct()`): Initializing Objects **25min**

6. 6. Destructors (`__destruct()`): Cleaning Up Objects **15min**

7. 7. Recap & Practical Task: Create Your First PHP Class and Object **40min**

Module 2: Module 2: OOP Principles: Encapsulation & Access Modifiers

5 Lessons

1. 8. Understanding Encapsulation: Bundling Data and Behavior **25min**
2. 9. Access Modifiers: `public`, `private`, `protected` **30min**
3. 10. Implementing Encapsulation with Getters and Setters (Accessor Methods) **35min**
4. 11. Property Visibility and Best Practices **20min**
5. 12. Recap & Practical Task: Encapsulate a User Class **45min**

Module 3: Module 3: OOP Principles: Inheritance

6 Lessons

1. 13. What is Inheritance? Extending Classes with `extends` **25min**
2. 14. Parent and Child Classes (Subclasses and Superclasses) **30min**
3. 15. Method Overriding: Customizing Parent Methods **25min**
4. 16. The `parent` Keyword: Accessing Parent Class Members **20min**
5. 17. The `final` Keyword for Classes and Methods (Preventing Overriding) **20min**
6. 18. Recap & Practical Task: Build a Simple Animal Class Hierarchy **50min**

Module 4: Module 4: OOP Principles: Polymorphism & Abstraction

6 Lessons

1. 19. Understanding Polymorphism: Many Forms, One Interface **25min**
2. 20. Abstract Classes and Abstract Methods: Defining Contracts **35min**
3. 21. Interfaces: Defining Behavior Blueprints **30min**
4. 22. Implementing Multiple Interfaces **20min**
5. 23. Type Hinting & Return Types for Robustness **25min**
6. 24. Recap & Practical Task: Implement Payment Gateways using Polymorphism and Interfaces **1hr**

Module 5: Module 5: Static Members, Constants, and Autoloading

6 Lessons

1. 25. Static Properties and Methods: Class-Level Members **25min**
2. 26. Class Constants: Defining Fixed Values within a Class **20min**
3. 27. The `self` Keyword vs. `static` Keyword (Late Static Binding) **25min**
4. 28. Introduction to Autoloading: Solving File Inclusion Problems **20min**
5. 29. Implementing Autoloading with `spl_autoload_register()` **30min**
6. 30. Recap & Practical Task: Create a Utility Class with Static Methods and Autoloading **50min**

Module 6: Module 6: Traits for Flexible Code Reusability

6 Lessons

1. 31. What are Traits? Overcoming PHP's Single Inheritance Limitation **25min**
2. 32. Defining and Using Traits in Classes **30min**
3. 33. Conflict Resolution with Traits (`insteadof`, `as`) **35min**
4. 34. Using Multiple Traits in a Single Class **20min**
5. 35. Traits within Traits **15min**
6. 36. Recap & Practical Task: Refactor Common Functionality into Traits **45min**

Module 7: Module 7: Magic Methods - Customizing Object Behavior

8 Lessons

1. 37. Introduction to Magic Methods: How They Work **20min**
2. 38. Property Overloading: `__get()`, `__set()`, `__isset()`, `__unset()` **35min**
3. 39. Method Overloading: `__call()`, `__callStatic()` **30min**
4. 40. Object Serialization: `__sleep()`, `__wakeup()` **25min**
5. 41. Object Conversion: `__toString()`, `__invoke()` **20min**
6. 42. Cloning Objects: `__clone()` **15min**
7. 43. Debugging Objects: `__debugInfo()` **15min**
8. 44. Recap & Practical Task: Create a Dynamic Configuration Object using Magic Methods **50min**

Module 8: Module 8: Namespaces and Exception Handling in OOP

7 Lessons

1. 45. Namespaces: Organizing Code and Preventing Name Collisions **25min**

2. 46. Declaring and Using Namespaces (`namespace`, `use`) **30min**

3. 47. Global Space and Importing Aliases **20min**

4. 48. Custom Exceptions in OOP: Creating Your Own Exception Classes **30min**

5. 49. Throwing and Catching Custom Exception Objects **25min**

6. 50. Finally Block and Nested Try/Catch **20min**

7. 51. Recap & Practical Task: Implement Namespaced Classes with Custom Exception Handling **50min**

Module 9: Module 9: Advanced OOP Principles & Design Patterns

14 Lessons

1. 52. SOLID Principles: An Overview for Robust Design **30min**

2. 53. Single Responsibility Principle (SRP) **20min**

3. 54. Open/Closed Principle (OCP) **20min**

4. 55. Liskov Substitution Principle (LSP) **20min**

5. 56. Interface Segregation Principle (ISP) **20min**

6. 57. Dependency Inversion Principle (DIP) **20min**

7. 58. Dependency Injection (DI): Concepts and Implementation **35min**

8. 59. Composition vs. Inheritance: When to Use Which **30min**

9. 60. Introduction to Common OOP Design Patterns **25min**

10. 61. Creational Patterns: Singleton Pattern **30min**

11. 62. Creational Patterns: Factory Pattern **25min**

12. 63. Structural Patterns: Adapter Pattern (Conceptual) **20min**

13. 64. Behavioral Patterns: Observer Pattern (Pub/Sub) **30min**

14. 65. Recap & Practical Task: Refactor a Small Project with SOLID Principles and DI **1hr 15min**

Module 10: Module 10: Final OOP Project: Building a PHP OOP Application

7 Lessons

1. 66. Project Overview: Planning an OOP-Based Application (e.g., Simple Blog/CMS) **1hr**

2. 67. Designing the Class Structure and Relationships **1hr 30min**

3. 68. Implementing Core Features with Encapsulation and Inheritance **2hr**

4. 69. Applying Interfaces and Abstract Classes for Flexibility **1hr 30min**

5. 70. Integrating Namespaces, Autoloading, and Custom Exceptions **1hr**

6. 71. Final Code Review and Best Practices for OOP in PHP **45min**

7. 72. Recap & Course Conclusion **40min**

This curriculum is subject to minor adjustments to ensure the most up-to-date and effective learning experience.