

INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Introduction to Object Oriented Programming

In this lesson, we will delve into the fundamental concepts of Object-Oriented Programming, a programming paradigm that is widely used in software development. OOP allows us to model real-world entities and their interactions in a more intuitive and organized manner. Let's start by understanding the key concepts of OOP.

Objectives of the Lesson

1. **Demonstrate Understanding of OOP Concepts:** By the end of this lesson, learners should be able to explain the core concepts of Object-Oriented Programming (OOP), including encapsulation, abstraction, inheritance, and polymorphism. They should be able to distinguish between classes and objects and understand how these concepts contribute to more organized and maintainable code.
2. **Create and Utilize Classes and Objects:** After studying this lesson, participants should be capable of creating their own classes and objects in PHP. They should be able to define properties, methods, setters, and getters within a class. Additionally, learners should be able to instantiate objects and apply the principles of encapsulation, inheritance, and polymorphism in their code.
3. **Evaluate Benefits of OOP in PHP Web Development:** As a result of this lesson, students should be able to identify the advantages of using Object-Oriented Programming in PHP web development. They should comprehend how OOP promotes modularity, reusability, and maintainability of code. Learners should also be able to explain how OOP principles, such as encapsulation and inheritance, contribute to the efficiency and scalability of web applications.

Tools needed

Text Editor (VS Code, Sublime Text, Brackets)

A text editor is an essential tool for writing and editing code. In the context of Object-Oriented Programming (OOP) and PHP, a good text editor provides a user-friendly interface for creating, modifying, and organizing your code files.

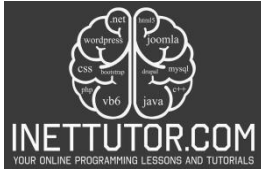
Visual Studio Code (VS Code): VS Code is a popular and powerful source code editor developed by Microsoft. It supports a wide range of programming languages, including PHP. It offers features like syntax highlighting, code completion, debugging, and version control integration through extensions. The versatility of VS Code makes it a great choice for OOP and PHP development, allowing you to streamline your coding workflow.

Sublime Text: Sublime Text is a lightweight yet feature-rich text editor known for its speed and simplicity. It provides customizable key bindings, a distraction-free mode, and a wide array of plugins that enhance its capabilities for PHP development. Sublime Text is particularly favored by developers who appreciate its performance and straightforward interface.

Brackets: Brackets is an open-source text editor designed specifically for web development, making it well-suited for working with HTML, CSS, and JavaScript, as well as PHP. It offers live preview functionality for web pages and includes features like inline editing and quick navigation. Brackets' focus on web-related technologies can be advantageous when working on PHP projects that involve web interfaces.

XAMPP is a cross-platform web server solution that simplifies the process of setting up a local development environment for PHP, MySQL, and Apache. The acronym XAMPP stands for "X" (cross-platform), "Apache," "MySQL," "PHP," and "Perl." It bundles these components along with other tools and utilities needed to create a complete server environment on your local machine.

XAMPP provides a user-friendly way to simulate a web server environment locally, allowing developers to test and develop PHP applications without the need for a remote server. It includes an Apache HTTP



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

server, MySQL database server, PHP interpreter, and more. This tool is particularly valuable for practicing PHP OOP concepts and creating web applications in a controlled environment before deploying them to a live server.

XAMPP's user interface simplifies the process of starting and stopping the services, managing databases, and configuring the server settings. It's an excellent tool for both beginners learning PHP and advanced developers working on larger projects.

What is Object Oriented Programming?

Object-Oriented Programming (OOP) is a programming paradigm that revolves around the concept of "objects". An object is a self-contained unit that combines both data and the methods (functions) that operate on that data. OOP helps us model complex systems by breaking them down into smaller, more manageable components.

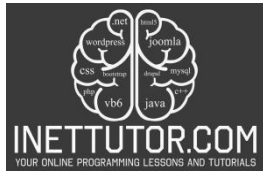
OOP is based on four main principles: encapsulation, abstraction, inheritance, and polymorphism. These principles provide a solid foundation for designing and structuring software.

Key Concepts of OOP

1. **Class:** A class is a blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that the objects created from the class will have.
2. **Object:** An object is an instance of a class. It represents a specific entity and encapsulates both data and methods related to that entity.
3. **Encapsulation:** Encapsulation is the practice of bundling data (attributes) and methods (functions) that operate on that data into a single unit (an object). It helps in controlling access to the internal details of an object, promoting data security and integrity.
4. **Abstraction:** Abstraction involves simplifying complex reality by modeling classes and objects that capture only the relevant attributes and behaviors. It hides the unnecessary details while emphasizing the essential aspects.
5. **Inheritance:** Inheritance is a mechanism that allows a new class (subclass or derived class) to inherit properties and behaviors from an existing class (base class or superclass). It promotes code reuse and enables the creation of hierarchical relationships between classes.
6. **Polymorphism:** Polymorphism allows objects of different classes to be treated as objects of a common superclass. It enables flexibility and extensibility by allowing different classes to implement their own versions of methods defined in a superclass.

Benefits of OOP

1. **Modularity:** OOP encourages the division of complex systems into smaller, self-contained modules (classes), making the codebase more manageable and maintainable.
2. **Reusability:** Inheritance and polymorphism facilitate code reuse, as common functionality can be defined in a base class and shared among multiple subclasses.
3. **Flexibility:** OOP enables developers to extend and modify existing code without affecting the entire system, promoting adaptability to changing requirements.
4. **Ease of Maintenance:** With clear class definitions and well-defined relationships, maintaining and debugging code becomes easier.
5. **Scalability:** OOP promotes a structured approach to software design, which is essential for building scalable and complex applications.



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

OOP in PHP

PHP (Hypertext Preprocessor) is a widely used server-side scripting language, especially suited for web development. It is known for its simplicity and flexibility in creating dynamic web pages and web applications. When combined with Object-Oriented Programming, PHP becomes a powerful tool for building complex and feature-rich web solutions.

What is a Class?

In PHP, a **class** is a blueprint or a template for creating objects. It defines the structure, properties, and behaviors that objects created from the class will possess. A class acts as a blueprint for creating instances (objects) that share the same attributes and behaviors.

A class consists of two main components:

1. **Properties (Attributes):** These are variables that hold data specific to an object. They define the characteristics or features of the object. Properties are often referred to as attributes or fields.
2. **Methods (Functions):** These are functions that define the behaviors or actions an object can perform. Methods are defined within the class and can access and manipulate the object's properties.

What is an Object?

An object is an instance of a class. It's a concrete entity created based on the blueprint provided by the class. In simpler terms, a class is like a recipe, and an object is the actual dish you create using that recipe. Objects are the runtime entities that can interact with each other and perform tasks based on the methods defined in the class.

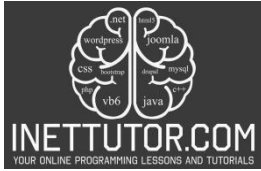
Using Object-Oriented Programming in PHP web development provides you with a structured and organized approach to building powerful applications. By embracing principles such as encapsulation, inheritance, and polymorphism, you can create maintainable, reusable, and scalable code. Remember that while OOP offers numerous benefits, it's important to choose the right design patterns and practices that best suit the needs of your project. With practice and experience, you'll become adept at crafting efficient and elegant web applications using PHP and OOP.

Source code example

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <?php
6. // Define a class named "Faculty"
7. class Faculty {
8.     // Properties (attributes)
9.     public $name;
10.    public $address;
11.    public $contact;
12.
13.    // Method to set the "name" property
14.    function set_name($name) {
15.        $this->name = $name;
16.    }
17.
18.    // Method to get the value of the "name" property
19.    function get_name() {
20.        return $this->name;
21.    }
22.
23.    // Method to set the "address" property
24.    function set_address($address) {
25.        $this->address = $address;
26.    }
27.
28.    // Method to get the value of the "address" property
29.    function get_address() {
30.        return $this->address;
31.    }
32.
33.    // Method to set the "contact" property
34.    function set_contact($contact) {
35.        $this->contact = $contact;
36.    }
37.
38.    // Method to get the value of the "contact" property
39.    function get_contact() {
40.        return $this->contact;
41.    }
42. }
43.
44. // Create an instance of the Faculty class
45. $my_info = new Faculty();
46.
47. // Set values for the properties of the object
48. $my_info->set_name('Your name');
49. $my_info->set_address('Your address');
50. $my_info->set_contact('Your contact');
51.
52. // Output the information using the object's methods
53. echo "Name: " . $my_info->get_name();
54. echo "<br>";
55. echo "Address: " . $my_info->get_address();
56. echo "<br>";
57. echo "Contact: " . $my_info->get_contact();
58. ?>
59.
60. </body>
61. </html>
```

Explanation:

1. The code starts with an HTML structure, indicating that this is a web page containing PHP code.
2. A PHP class named Faculty is defined. It has three properties: \$name, \$address, and \$contact.



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

3. For each property, there are two methods defined: a setter method (e.g., `set_name()`) to set the value of the property, and a getter method (e.g., `get_name()`) to retrieve the value of the property.
4. An instance of the Faculty class is created using the `$my_info` variable.
5. The `set_name()`, `set_address()`, and `set_contact()` methods are called to set the values of the properties for the `$my_info` object.
6. The `get_name()`, `get_address()`, and `get_contact()` methods are used to retrieve the values of the properties, and these values are displayed on the web page using the echo statements.

Output:



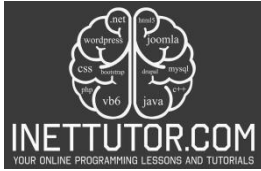
Name: Your name
Address: Your address
Contact: Your contact

Summary

This blog post's highlighted the essence of Object-Oriented Programming and how it integrates with PHP without any issues. You are prepared to start coding in a way that prioritizes code organization, reusability, and maintainability if you have a basic understanding of OOP's principles and PHP's capabilities. Understanding the power of OOP and PHP will surely advance your programming skills, whether you're a beginner or a seasoned developer. Create beautiful, effective, and scalable applications by combining PHP and Object-Oriented Programming, the dynamic pair!

Quiz

1. What is Object-Oriented Programming (OOP)?
 - a) A programming language
 - b) A programming paradigm
 - c) A data structure
 - d) An operating system
2. Which OOP principle emphasizes bundling data and methods into a single unit?
 - a) Inheritance
 - b) Encapsulation
 - c) Polymorphism
 - d) Abstraction
3. Which OOP principle focuses on simplifying complex realities by hiding unnecessary details?
 - a) Abstraction
 - b) Encapsulation
 - c) Inheritance
 - d) Polymorphism



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

4. In OOP, what is a class?
 - a) An instance of an object
 - b) A property of an object
 - c) A blueprint for creating objects
 - d) A method of an object

5. What do objects represent in OOP?
 - a) Blueprints for classes
 - b) Methods of a class
 - c) Properties of a class
 - d) Instances of a class

6. Which term refers to creating a new object based on a class?
 - a) Defining
 - b) Initializing
 - c) Instantiating
 - d) Constructing

7. What does the term "inheritance" mean in OOP?
 - a) A process of copying data from one object to another
 - b) A way to reuse and extend properties and methods of an existing class
 - c) The ability of an object to access properties of another object
 - d) A technique for encapsulating data within a class

8. What is the main advantage of using polymorphism in OOP?
 - a) It allows objects to inherit properties from a superclass
 - b) It promotes code reuse through the use of interfaces
 - c) It enables objects of different classes to be treated as objects of a common superclass
 - d) It simplifies the process of creating new classes

9. How does OOP enhance code organization and maintainability?
 - a) By using only primitive data types
 - b) By avoiding the use of methods
 - c) By breaking down applications into smaller, reusable components
 - d) By eliminating the need for classes and objects

10. Which programming language is widely used for web development and supports OOP principles?
 - a) JavaScript
 - b) Python
 - c) Ruby
 - d) PHP