Course Outline: Object-Oriented Programming with PHP

**Prelim**

**Week 1: Introduction to Object-Oriented Programming in PHP**

* Introduction to programming paradigms: Procedural vs. Object-Oriented
* Core OOP concepts:
  + Classes and objects
  + Properties and methods
  + Encapsulation
  + Inheritance
  + Polymorphism
  + Abstraction
* Benefits of OOP: Modularity, reusability, maintainability
* Introduction to PHP classes:
  + Defining classes (class keyword)
  + Creating objects (instances)
  + The \_\_construct method (constructor)
  + Instance vs. static properties
  + Using $this keyword

**Week 2: Classes and Objects in Detail**

* Class definition syntax in PHP
* Properties (attributes):
  + Defining and accessing properties
  + Initializing properties with constructor
  + Public, protected, and private properties
* Methods:
  + Defining methods
  + $this usage
  + Calling methods
  + Instance methods and static methods (static keyword)
* Object lifecycle:
  + Object creation
  + Object destruction and \_\_destruct

**Week 3: Encapsulation**

* Concept of encapsulation in PHP
* Data hiding and access control
* Attribute visibility:
  + Public
  + Protected
  + Private
* Getter and setter methods:
  + Creating getters for private properties
  + Creating setters for private properties
* Benefits of encapsulation:
  + Data protection
  + Code organization
  + Reduced complexity

**Midterm**

**Week 4: Inheritance**

* Concept of inheritance
* Base (parent) and derived (child) classes
* "is-a" relationship
* Types of inheritance:
  + Single inheritance
  + Hierarchical inheritance
  + Multilevel inheritance
* Method overriding in PHP
* Using parent:: keyword
* Inheritance with constructors

**Week 5: Polymorphism**

* Concept of polymorphism
* Method overriding
* Interface-based polymorphism (interface keyword)
* Abstract methods for polymorphism
* Late static binding (static::)

**Week 6: Abstract Classes and Interfaces**

* Abstract classes (abstract keyword)
* Abstract methods
* Purpose of abstract classes:
  + Template for subclasses
  + Enforcing method implementation
* Interfaces:
  + Defining contracts
  + Multiple interface implementation
* Implementing abstract classes and interfaces

**Semi-Final**

**Week 7: Special Methods and Magic Methods in PHP**

* Introduction to magic methods
* Common magic methods:
  + \_\_construct, \_\_destruct
  + \_\_get, \_\_set
  + \_\_toString, \_\_invoke
  + \_\_call, \_\_callStatic
  + \_\_clone
* Implementing magic methods to enhance functionality

**Week 8: Exception Handling in OOP with PHP**

* Exception handling in PHP:
  + try, catch, finally blocks
  + Throwing exceptions (throw)
* Custom exceptions:
  + Creating classes extending Exception
* Exception handling in method calls
* Best practices for exception usage

**Week 9: Namespaces, Autoloading & Composer**

* Introduction to namespaces (namespace keyword)
* Benefits of namespaces in large projects
* Using use statements to import classes
* Autoloading classes with spl\_autoload\_register
* PSR-4 autoloading standard
* Using Composer for autoloading
* Organizing project directory structure for maintainability
* Preparing code for use in PHP frameworks

**Final**

**Week 10: PHP Handling**

* PHP form handling with OOP
  + Classes for form processing
  + Handling $\_POST and $\_GET
  + File uploads with OOP ($\_FILES)
* Session and cookie handling in OOP
  + Session handler classes
  + Cookie management methods
* File handling in OOP
  + Reading and writing files (fopen, fwrite, fread)
  + File handling classes
* Database handling with PDO in OOP
  + Database connection classes
  + CRUD operations with prepared statements

**Weeks 11 – 12: Project Development**

* Students work on an OOP PHP-based project
* Individual or group work
* Must demonstrate OOP concepts and PHP handling
* Example projects:
  + Student Management System
  + Inventory Management System
  + Online Booking System
* Focus on:
  + Good OOP design
  + Proper project structure
  + PHP handling for forms, sessions, and databases
  + Error handling
  + Documentation