**Week 1: PHP Functions**

1. **Introduction to PHP Functions**
   * Syntax and definition (function keyword)
   * Parameters and arguments (default parameters, passing by value vs. reference)
   * Return values (single and multiple returns using arrays)
2. **Function Scope and Global Variables**
   * Local vs. global variables
   * Using the global keyword and the $GLOBALS array
3. **Advanced Functions**
   * Variable-length argument lists (...$args)
   * Anonymous functions (Closures)
   * Callback functions
   * Recursive functions
4. **Best Practices**
   * Writing reusable functions
   * Avoiding side effects
   * Proper naming conventions
   * Documentation using PHPDoc

**Week 2: Introduction to OOP in PHP**

1. **OOP Concepts Overview**
   * Classes and Objects
   * Properties and Methods
   * Constructors and Destructors (\_\_construct(), \_\_destruct())
2. **Encapsulation**
   * Public, Private, and Protected visibility
   * Getters and Setters
3. **Static Properties and Methods**
   * The static keyword
   * Accessing static properties and methods
4. **The this and self Keywords**
   * Difference between $this and self::

**Week 3: Inheritance and Polymorphism**

1. **Inheritance**
   * Creating subclasses using the extends keyword
   * Overriding methods
   * Calling parent methods (parent::)
2. **Polymorphism**
   * Abstract classes and methods
   * Interfaces and implements
   * Type hinting and return types
3. **Final Classes and Methods**
   * Using the final keyword to prevent inheritance or overriding

**Week 4: Advanced OOP Concepts**

1. **Traits**
   * What are Traits and why use them?
   * Defining and using traits (use keyword)
   * Resolving conflicts between traits
2. **Namespaces**
   * What are namespaces?
   * Declaring and using namespaces (namespace, use)
3. **Dependency Injection (DI)**
   * Concept of DI
   * Constructor injection vs. Setter injection
   * Benefits of DI in maintaining loosely coupled code
4. **Design Patterns (Introduction)**
   * Singleton pattern
   * Factory pattern
   * MVC pattern

**Week 5: Error Handling and Testing**

1. **Error Handling in PHP**
   * Exceptions and try-catch blocks
   * Custom Exception classes
2. **Unit Testing**
   * Introduction to testing in PHP using PHPUnit
   * Writing unit tests for functions and OOP classes
3. **Error Logging**
   * Built-in error handling and logging functions (error\_log, set\_error\_handler, set\_exception\_handler)

**Week 6: Real-World Application**

1. **Project: Building a Simple OOP-based Application**
   * Create a small project (e.g., a Product Management System) that implements:
     + CRUD operations
     + Object-Oriented Design principles
     + Proper validation and error handling
2. **Refactoring and Optimization**
   * Refactoring code for better readability and maintainability
   * Optimization techniques (e.g., caching, minimizing resource usage)