Object Oriented Programming

For week 10, of our comprehensive training program on Object-Oriented Programming (OOP) we will delve into the fundamental concepts that underpin modern software development, focusing on the principles of OOP, including classes, objects, inheritance, polymorphism, and encapsulation. By the end of this week, you will have a solid understanding of how to create robust, scalable, and maintainable code using OOP principles, as well as the ability to apply these concepts to solve real-world problems efficiently.

Software Development Week 10

Learning Objectives for the week

At the end of this week you should be able to;

- **Explain the key concepts of Object-Oriented Programming (OOP), including classes, objects, inheritance, polymorphism, and encapsulation.**
- **Create and use classes and objects to model real-world entities.**
- **Implement encapsulation to protect and control access to class data.**
- **Develop class hierarchies using inheritance for code reuse and flexibility.**
- **Apply polymorphism through method overloading and overriding.**
- **Design and implement scalable, maintainable solutions using OOP principles.**

Software Development Week 10

Online Learning Modules

This is the online module that you have to complete this week.

- Introduction to Object Oriented Programming Full Course: Click Here 1
- Object Oriented Programming in Javascript: Click Here 2

Additional Resource links
1. Click here - Text
2. Article - Text
3. Download - Ebook

Software Development Week 10

Weekly Applied Learning Assignment

Objective:

Design and implement a simple library management system that demonstrates your understanding of key OOP principles: classes, objects, inheritance, polymorphism, and encapsulation.

Requirements:

1. **Create the following classes:**
    - **Book**: Attributes like title, author, ISBN, and availability status.
    - **Member**: Attributes like name, member ID, and borrowed books.
    - **Librarian**: Inherits from Member and has additional responsibilities like adding/removing books and checking the availability of books.
2. **Encapsulation:**
    - Ensure that book availability and member details are properly encapsulated with getter and setter methods.
3. **Inheritance:**
    - Use inheritance to create different types of users (e.g., Librarian and Member), where the Librarian can have additional privileges over a regular Member.
4. **Polymorphism:**
    - Demonstrate polymorphism by creating a method for borrowing books that behaves differently for Librarian and Member.
5. **Optional Feature:**
    - Implement a simple search feature that allows members to search for books by title or author.
    - Provide a simple demo showcasing the system's functionality.

Submission:

- **Submit your code and summary via CodePen.**