

## Week 4 Exercises:

Here are five assignment questions and a project for the past week's topic on Object-Oriented Programming (OOP):

### 1. Class and Object Basics:

Define a class `Student` with attributes `name` and `age`. Create an object of the class and print out the student's name and age.

### 2. Inheritance and Polymorphism:

Create a base class `Shape` with a method `area()`. Derive two classes `Circle` and `Rectangle` from `Shape` and implement the `area()` method for each. Demonstrate polymorphism by calculating and comparing areas of a circle and a rectangle.

### 3. Encapsulation and Methods:

Create a class `BankAccount` with attributes `account\_number`, `balance`, and private attribute `pin`. Implement methods `deposit(amount)` and `withdraw(amount)` that modify the balance. Add a method `check\_balance(pin)` that returns the balance if the correct PIN is provided.

### 4. Class Composition:

Define a class `Author` with attributes `name` and `birth\_year`. Create a class `Book` with attributes `title`, `publication\_year`, and an `Author` object. Implement a method to print out the book's details, including the author's name and birth year.

### 5. Method Overriding:

Create a base class `Animal` with a method `make\_sound()`. Derive two classes `Dog` and `Cat` from `Animal` and override the `make\_sound()` method. Write a function that takes an `Animal` object as an argument and calls its `make\_sound()` method.

## Weekly Project: Simple Library System

Design a simple library system using Object-Oriented Programming (OOP) principles. Your program should have the following features:

- Create a class `Book` with attributes `title`, `author`, and `available\_copies`.
- Create a class `Library` that manages a collection of `Book` objects.
- Implement methods to add a new book, borrow a book (reduce available copies), and return a book (increase available copies).
- Include error handling to prevent borrowing more copies than available or returning books that were not borrowed.

- Create a few instances of `Book` and `Library` classes and demonstrate various operations.

Remember to follow OOP principles such as encapsulation, inheritance, and polymorphism in your project. This project will provide hands-on experience in applying OOP concepts to create a real-world application.

Submit your exercises via email to the course instructor.