

## Homework Assignment for Lesson 3: Encapsulation and Abstraction

### Problem 1: Encapsulation and Getter/Setter Methods

- 1.1. Define a class called `BankAccount` with the following attributes:
  - `account_number` (string)
  - `balance` (float)
- 1.2. Encapsulate the `balance` attribute by making it private. Create getter and setter methods (`get_balance` and `set_balance`) to access and modify the balance.
- 1.3. Implement a method called `withdraw` that takes an amount as an argument and updates the balance accordingly. Ensure that the balance cannot go below zero.

### Problem 2: Abstraction and Abstract Class

- 2.1. Define an abstract class called `Vehicle` using the `ABC` module. Include the following abstract methods:
  - `start_engine`
  - `stop_engine`
  - `drive`
- 2.2. Create a concrete subclass `Car` that inherits from the `Vehicle` class. Implement the abstract methods with appropriate messages for starting, stopping, and driving the car.
- 2.3. Instantiate a `Car` object and call each of the implemented methods.

### Problem 3: Application of Encapsulation and Abstraction

- 3.1. Think of a real-world scenario where encapsulation and abstraction could be applied. Describe the scenario, including a class with encapsulated attributes and abstract methods.
- 3.2. Write Python code to implement the described scenario. Instantiate an object of the class and demonstrate the use of encapsulation and abstraction.