Problems based on Object Oriented Concepts

1. You're developing a library management system. You need to represent different types of library items. All items have a `title`, `author`, and `publication_year`. There are different types of items: `Book`, `DVD`, and `Magazine`. `Book` has an `isbn` and `number_of_pages`. `DVD` has a `running_time` and `genre`. `Magazine` has an `issue_number` and `publisher`.

    - **Question 1:** Design a class hierarchy for these library items. Identify the base class and the subclasses. What attributes would each class have?
    - **Question 2:** Implement the classes in Ruby. Include methods to access and modify the attributes.
    - **Question 3:** Create instances of different library items (a Book, a DVD, and a Magazine). Demonstrate how you would access their specific attributes.
    - **Question 4:** Add a method `due_date` to the base class. This method should calculate the due date based on the current date and a loan period (you can assume a default loan period).
    - **Question 5:** The `due_date` calculation is different for DVDs (shorter loan period). How would you handle this difference in your code? Implement the specific `due_date` calculation for DVDs. Explain the concept you used.

2. You're building an e-commerce platform. You need to represent different types of products. All products have common attributes like `name`, `price`, and `description`. However, some products have specific attributes. Books have an `author` and `isbn`, while clothing items have a `size` and `color`.

    - **Question 1:** Design a class hierarchy using Ruby to represent these products. Identify the base class and any subclasses you would create. What attributes would each class have?
    - **Question 2:** Implement the classes you designed in Question 1. Include methods to access and modify the attributes.
    - **Question 3:** Create instances of different product types (e.g., a book and a clothing item). Demonstrate how you would access their attributes.
    - **Question 4:** You need to add a `discount` method to all products. Where would you define this method in your class hierarchy, and why? Implement the method.
    - **Question 5:** Books have a different discount calculation than clothing. How would you handle this difference in your code? Implement the specific discount calculations for books and clothing. Explain the concept you used.

3. You're building a game with different types of characters: heroes, villains, and monsters. All characters need to have certain abilities, such as the ability to move,

attack, and defend. However, some characters might have additional, unique abilities. You want to avoid code duplication and keep your classes organized.

- **Question 1:** How can you use mixins to represent the common abilities (move, attack, defend) that all characters share? Design a mixin (or mixins) that encapsulates these behaviors.
- **Question 2:** Show how you would include this mixin in your `Hero`, `Villain`, and `Monster` classes.
- **Question 3:** Suppose heroes have a special "healing" ability, and villains have a "poison" ability. How can you use mixins to represent these unique abilities without affecting the other character types?
- **Question 4:** You want to add a "level up" mechanism to your game. This mechanism should affect the character's attack and defense abilities. How can you use a mixin to encapsulate the level-up logic so that it can be applied to any character?