Exercise 1: BankAccount with Private Attributes

Create a class called `BankAccount`.

Instructions:

- Create a class named `BankAccount`
- Define an `__init__` method with parameters `owner_name` and `initial_balance`
- Store the balance as a **private attribute** (use `_balance`)
- Create a method `deposit(amount)` that increases the balance only if `amount` is positive
- Create a method `withdraw(amount)` that checks if the balance is sufficient before withdrawing
- Create a method `get_balance()` that returns the current balance
- In the main program, create an account and interact with it only through methods (no direct access to `_balance`)

Exercise 2: Student and Classroom Interaction

Create two classes: `Student` and `Classroom`.

Instructions:

- Create a class `Student` with:
  - an `__init__` method taking `name`
  - an attribute `grades` initialized as an empty list
  - a method `add_grade(grade)`
  - a method `average()` that returns the student’s average grade
- Create a class `Classroom` with:
  - an `__init__` method taking a classroom name
  - an attribute `students` initialized as an empty list
  - a method `add_student(student)` that adds a `Student` object
  - a method `class_average()` that computes the average of all students
- Create several students, add grades, add them to a classroom, and display results

---

Exercise 3: Car with State and Validation

Create a class called `Car`.

Instructions:

- Create a class `Car`
- Define an `__init__` method with parameters `brand`
- Add attributes `speed` (initially 0) and `engine_on` (initially False)
- Create a method `start_engine()` that turns the engine on
- Create a method `stop_engine()` that stops the engine and sets speed to 0
- Create a method `accelerate(value)` that increases speed **only if the engine is on**
- Create a method `brake(value)` that decreases speed but never below 0
- In the main program, test incorrect and correct usage (e.g. accelerate before starting)

Exercise 4: Library and Book Management

Create two classes: `Book` and `Library`.

Instructions:

- Create a class `Book` with:
    - an `__init__` method taking `title` and `author`
    - an attribute `is_borrowed` initialized to False
- Create a class `Library` with:
    - an `__init__` method taking a library name
    - an attribute `books` initialized as an empty list
    - a method `add_book(book)`
    - a method `borrow_book(title)` that marks a book as borrowed if available
    - a method `return_book(title)` that marks a book as available again
- Use objects of type `Book` inside the `Library` class

Exercise 5: Class Attribute and Object Counter

Create a class called `User`.

Instructions:

- Create a class `User`
- Define a **class attribute** `user_count` initialized to 0

- In the `__init__` method, increment `user_count` every time a new object is created
- Add attributes `username` and `email`
- Create a method `display_info()` that shows the user’s information
- Create a **class method** that returns the total number of users created
- Create several users and display the total count