

Coding Exercises

Q1

Create a class **BankAccount** with a private field `_balance`.

- Add a getter `balance` that returns the balance.
- Add a setter `balance` that prevents setting it to a negative value (print 'Invalid balance' if attempted).
- In `main()`, demonstrate creating an account, updating the balance, and trying to set a negative balance.

Q2

Create a class **Car** with private fields `_brand` and `_year`.

- Add setters that reject empty brand names and years less than 1886 (first car invention).
- Add getters for both.
- In `main()`, demonstrate creating two car objects (one valid, one invalid input).

Q3

Create a class **Grade** with a private field `_score`.

- The setter should only accept values 0–100, otherwise print 'Invalid score'.
- Add a getter and a computed getter `isPass` that returns true if `score ≥ 50`.
- In `main()`, demonstrate updating the score multiple times and printing results.

Q4

Create a class **Product** with private fields `_name` and `_price`.

- Reject empty names and negative prices in setters.
- Add a computed getter `discountedPrice` that returns the price with a 10% discount applied.
- In `main()`, demonstrate setting values and printing the original and discounted price.

Q5

Create a class **Book** with private fields `_title` and `_pages`.

- Add setters: reject empty titles and `pages ≤ 0`.
- Add a getter `title` and a computed getter `readingTime` that assumes 2 minutes per page.
- In `main()`, create a book, print its title and estimated reading time.

Q6

Given a string `s` containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is **valid**.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.

2. Open brackets must be closed in the correct order.
3. Every close bracket has a corresponding open bracket of the same type.

Examples:

- '()' → Valid
- '()[]{}' → Valid
- '[]' → Invalid
- '([)]' → Invalid
- '{}[]' → Valid

Q7

Ask the user to input a list of integers.

- Print the **largest number**, the **smallest number**, and their **difference**.
- Calculate the **average** of the list.
- Print all numbers that are **above the average**.
- Finally, print how many numbers are **even** and how many are **odd** in the list.