Lab 08 Tasks

Task 01

Create two classes, **Temperature** and **Humidity**, that store temperature (in °C) and humidity (in %), respectively. Declare a friend function named calculateHeatIndex() that computes a simplified heat index using both classes' private data.

Requirements:

- Use a friend function to access private members from both classes.
- Include a main function that instantiates objects with sample values (e.g., 32°C and 70%) and prints the calculated heat index.

Task 02

Design a **Book** class that holds details such as title and price, and a **Librarian** (or Manager) class that can view and modify these details. Make the Librarian a friend class of Book so that it can apply discounts.

Requirements:

- Implement a friend class declaration in Book.
- In Librarian, add functions to display book details and apply a discount (modify the price).
- In main(), create a Book object, display its details, apply a discount, and display the updated details.

Task 03

Implement a **Student** class that stores a student's name and an array of three grades. Also, create a **Teacher** class (declared as a friend in Student) that can update and view these grades. Additionally, include a friend function to compute the average grade.

Requirements:

- Use friend class declaration so Teacher can access and modify Student's grades.
- Define a friend function calculateAverageGrade() to compute the average.
- In main(), simulate a grading scenario by displaying grades, updating one grade, and showing the new average.

Task 04

Build a **Currency** class (similar to the provided Rupee example) that stores a monetary value. Overload the following operators:

- Unary minus (to negate the currency amount)
- Addition (+) to add two currency values
- Subtraction (–) to subtract one currency from another
- Addition assignment (+=) and subtraction assignment (–=)
    - Also, overload the stream insertion operator (<<) for easy output.

      Requirements:
    - Ensure that the overloaded operators perform arithmetic on the stored value.
    - Write a main() function to demonstrate the usage by showing addition, subtraction, and applying discounts.

Task 05

Develop a **Fraction** class that represents a mathematical fraction (with numerator and denominator). Overload the arithmetic operators (+, -, *, /) so that fractions can be added, subtracted, multiplied, and divided. Additionally, overload the stream insertion operator (<<) for formatted output (e.g., “3/4”).

Requirements:

- Implement operator functions for the four arithmetic operations.
- Validate that denominators are non-zero (you can assume valid input for simplicity).
- Write a main() function to create a few Fraction objects, perform arithmetic operations, and output the results.