OOP with C++ Homework 1

C++ Homework Assignment: Vending Machine Simulation with Admin and User Modes

Objective:

- Write a C++ program that simulates a vending machine with two modes: User (selling mode) and Admin.
- The vending machine will allow users to select items and make payments, while the admin will have additional capabilities to manage stock, refill change, and collect money.
- The program must use SQLite to store and manage item stock, payment records, and change collections.
- The system should stop operating automatically if certain conditions are met, and users will only be able to exit the program.

Requirements:

1. Program Modes:
    - User Mode (Selling Mode):
        - Users can select an item from the menu by entering its unique selection code and pay using various denominations (100 THB, 20 THB, 10 THB, 5 THB, 1 THB).
        - After each purchase, decrease the stock of the selected item by 1.
        - The program will check if the item is in stock and calculate the total payment.
        - If the payment is insufficient, the user will be prompted to insert additional money.
        - If the user has completed the payment, the program will return the change and update the stock.
        - Out of Stock Handling: If a product is out of stock, display a sign (such as "OUT OF STOCK") to indicate that the item is unavailable for purchase.
        - Stop operation: The vending machine will stop selling automatically if the following conditions are met:
            - 50% or more of the product categories are out of stock.
            - There is no money in the change box (even if one denomination is missing).
            - The coin collection box is full (even if one domination is full).
    - Admin Mode: Admins can manage the vending machine operations, including:
        - Set initial stock: Admin can set the initial stock for each item.
        - Re-stock: Admin can refill the stock of any item.
        - Check money change and money collection box: Admin can check how much money is in the change box and the collection box (separate records).
        - Collect money: Admin can collect money from the collection box and refill the change box.
2. Database:
    - Use SQLite to store data.
    - The program should create an SQLite database table with the format stock_<student_id>, where <student_id> is your unique student ID number (e.g., stock_12345).
      Example:
        - Stock_12345 table will store the vending machine data for the student with ID 12345.
    - Provide a supplementary program to create the necessary tables in the SQLite database.

Submission Guidelines:

1. Submit the C++ program as a .cpp file that fulfills the requirements outlined above.
2. Submit the SQLite schema creation program and database schema file that sets up the database tables.-
3. Ensure that your code compiles and runs correctly with appropriate error handling.

Evaluation Criteria:

1. Correctness: Does the program fulfill all functionality requirements (both user and admin modes)?
2. Database Integration: Does the program interact with SQLite correctly to manage stock, payments, and money?
3. Error Handling: Are invalid inputs (item code, denominations, etc.) handled properly?
4. Code Structure: Is the code well-structured, readable, and well-commented?
5. Program Flow: Does the program flow logically between user and admin modes, with clear prompts and actions?

Good luck!

Submission: within Jan 15, 2025 19:00
https://docs.google.com/forms/d/e/1FAIpQLSd_CZOwrSd_qlmRUh7zSW-KmXE03VgThZDucGeZ7OTJgTYseA/viewform?usp=header