LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0023**
LOC:	**500**
Slot(s):	**N/A**

Title

GYM Management

Background

A fitness center requires a comprehensive program to efficiently manage member registrations, class schedules, equipment inventory, and financial transactions within its facility. The system aims to streamline member management, class scheduling, equipment maintenance, and overall gym management.

Program Specifications

Build a gym management program with the following basic functions:

1. **Member Management:**
   - Create a new member
   - Sort and print the list of members ascending by name
   - View and update existing member information:
     - Search member by member id.
     - Edit member information: name, phone number, address, etc.
   - Delete a member:
     - Remove a member from the list upon confirmation.
2. **Equipment Management:**
   - Add new equipment
   - Sort and print the equipment list descending by name.
   - Update and manage equipment inventory
   - Remove equipment
3. **Class Management:**
   - Create a new class
   - Update class information.
   - Remove a class
4. **Reporting and Analytics:**
   - Generate reports:
     - Analyze popular equipment and class occupancy.
     - Provide insights on class preferences and revenue generated.
5. **Data Management**

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

Features:

- **Function 1: Create a new member - 50 LOC**
    - Collect member details: ID, name, address, contact information, membership type, etc.
    - Validate input data for completeness and correctness.
    - Add the new member to the member test file.
- **Function 2: Sort and display existing member information - 25 LOC**
    - Sort the list of members ascending by name
    - Print the list of members after sorting.
- **Function 3: View and update existing member information - 50 LOC**
    - Search members by membership ID.
    - Edit member details: modify personal information.
- **Function 4: Delete a member – 50 LOC**
    - Search members by membership ID.
    - Remove a member from the system upon confirmation.
- **Function 5: Add new equipment– 50 LOC**
    - Input equipment details: equipment ID, name, type, quantity, condition, etc.
    - Ensure data integrity and uniqueness.
    - Store equipment information in the text file.
- **Function 6: Sort and display equipment - 25 LOC**
    - Search equipment by equipment id.
    - Modify equipment information: update name, quantity, or edit details.
- **Function 6: Update and manage equipment - 50 LOC**
    - Sort the list of equipment ascending by name
    - Print the list of equipment after sorting.
- **Function 7: Remove equipment - 25 LOC**
    - Search equipment by equipment id.
    - Delete equipment from the system after confirmation, considering current usage.
- **Function 8: Add new classes - 50 LOC**
    - Input class details: class ID, name, schedule, capacity, member, equipment, etc.
    - Ensure data integrity and uniqueness.
    - Store class information in the text file.
- **Function 8: Update and manage class - 50 LOC**
    - Search classes by class id.
    - Modify class information: update schedule, change members, or adjust equipment.
- **Function 9: Remove a class - 25 LOC**
    - Search classes by class id.
    - Delete a class from the system after confirmation, considering enrolled members.
- **Function 10: Data Management – 50 LOC**
    - ***Data persistence:***
        - Save member and class information to the binary file (e.g .dat, .bin).
        - Ensure consistent data storage and retrieval.
    - ***User Interface:***
        - Create an intuitive and user-friendly interface.
        - Incorporate menus and prompts for easy navigation between functionalities.

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

***Note: each member can have joined in many classes.**

The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.

The lecturer will explain the requirement only once in the first slot of the assignment.