

BÀI TẬP LỚN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

1. Đề tài 1: Xây dựng chương trình quản lý Đặt phòng Khách sạn

Hãy sử dụng các kiến thức về lập trình hướng đối tượng C++ để xây dựng chương trình hỗ trợ nhân viên Lễ tân quản lý trạng thái phòng và khách lưu trú.

Yêu cầu chức năng cơ bản:

Hệ thống Menu và Đăng nhập (0.5đ):

- Tạo 1 danh sách tài khoản lễ tân có sẵn (Username, Password).
- Yêu cầu đăng nhập trước khi vào menu chính.

Quản lý Phòng (1đ):

- Thông tin mỗi phòng gồm: Mã phòng (VD: P101), Loại phòng (Đơn, Đôi, VIP), Giá phòng/đêm, Trạng thái (Trống / Đang có khách / Đang dọn dẹp).
- Tạo danh sách quản lý các phòng hiện có trong khách sạn.
- Các thao tác: Thêm phòng mới, Cập nhật trạng thái phòng, Cập nhật giá, Xóa phòng, Tìm kiếm phòng theo mã hoặc theo trạng thái (VD: Tìm tất cả phòng đang "Trống").

Quản lý Khách hàng (1đ):

- Thông tin khách hàng: Mã khách hàng (CCCD/Passport), Họ tên, Số điện thoại, Năm sinh.
- Các thao tác: Thêm khách hàng mới, Cập nhật thông tin khách, Tìm kiếm khách hàng.

Quản lý Đặt phòng & Trả phòng (3đ):

- Thông tin một lượt đặt phòng (Booking) gồm: Mã booking, Mã khách hàng, Mã phòng, Ngày check-in (giờ, ngày, tháng, năm), Ngày check-out dự kiến.
- Quy trình Check-in (Đặt phòng):
 - Kiểm tra trạng thái phòng. Chỉ cho phép đặt nếu phòng đang ở trạng thái "Trống".
 - Sau khi đặt thành công, hệ thống tự động cập nhật trạng thái phòng sang "Đang có khách".
- Quy trình Check-out (Trả phòng):
 - Nhập Mã phòng hoặc Mã khách hàng để tìm thông tin đặt phòng.
 - Tính toán tổng tiền dựa trên: (Thời gian hiện tại - Thời gian check-in) * Giá phòng (có thể đơn giản là tính số ngày cho dễ).
 - Cập nhật trạng thái phòng về "Đang dọn dẹp" hoặc "Trống".
 - Lưu thông tin vào lịch sử doanh thu.

Báo cáo & Lịch sử (1.5đ):

- Tạo danh sách lưu trữ lịch sử các giao dịch đã hoàn tất (đã check-out).
- Thông tin lưu: Mã khách, Mã phòng, Ngày đến, Ngày đi, Tổng tiền thanh toán.

- Chức năng: Tra cứu lịch sử lưu trú theo Mã khách hàng (để biết khách này từng đến bao nhiêu lần, tổng tiền đã chi).

Yêu cầu các chức năng mở rộng:

Lưu trữ dữ liệu (1đ):

- Sử dụng file text để lưu trữ dữ liệu bền vững (bản chất giống như dùng file thay thế cho database), thay vì chỉ lưu vào vector mỗi khi khởi động chương trình là mất dữ liệu lần chạy trước -> không được tính điểm phần này.
- Khi chương trình bắt đầu (Start), đọc dữ liệu từ file vào các vector/List. Khi chương trình kết thúc (Exit) hoặc sau mỗi giao dịch quan trọng, ghi đè lại dữ liệu từ vector xuống file để đảm bảo không mất dữ liệu.

Quản lý ngoại lệ (Exception Handling) (1đ):

- Xây dựng các lớp ngoại lệ (custom exception) kế thừa từ `std::runtime_error`:
 - `RoomNotAvailableException`: Ném ra khi cố đặt một phòng đang có khách hoặc đang bảo trì.
 - `InvalidCustomerInfoException`: Khi số điện thoại hoặc CCCD không đúng định dạng.
 - `AuthenticationFailedException`: Khi đăng nhập sai quá 3 lần.
 - Sử dụng khối try-catch tại hàm main hoặc các hàm xử lý menu để bắt lỗi và thông báo thân thiện, không để chương trình bị crash.

Template Class (Generic Programming) (1đ):

- Xây dựng một lớp Template tên là `QuanLy<T>` (Manager).
- Lớp này sẽ chứa một `std::vector<T>` (hoặc cấu trúc danh sách tự định nghĩa) và chịu trách nhiệm cho các thao tác chung:
 - `Them(T item)`: Thêm đối tượng vào danh sách.
 - `Xoa(string id)`: Xóa đối tượng theo ID.
 - `TimKiem(string id)`: Trả về đối tượng hoặc chỉ số.
 - `HienThiDanhSach()`: In toàn bộ danh sách.
- Áp dụng lớp `QuanLy<T>` này để quản lý: `QuanLy<Phong>`, `QuanLy<KhachHang>`, `QuanLy<TaiKhoan>`, `QuanLy<LichSuDatPhong>`.

2. Đề tài 2: Xây dựng chương trình quản lý Cho thuê xe Ô tô

Bạn hãy sử dụng các kiến thức về lập trình hướng đối tượng C++ để xây dựng chương trình hỗ trợ nhân viên cửa hàng quản lý danh sách xe ô tô và các hợp đồng cho thuê.

Yêu cầu chức năng cơ bản:

Hệ thống Menu và Đăng nhập (0.5đ):

- Tạo danh sách tài khoản quản trị viên (Username, Password).
- Yêu cầu đăng nhập hợp lệ trước khi truy cập vào menu chức năng chính.

Quản lý Ô tô (1đ):

- Thông tin mỗi xe gồm: Biển số xe (Mã định danh duy nhất), Tên xe (VD: Toyota Vios, VinFast VF8), Loại xe (4 chỗ, 7 chỗ, Bán tải), Giá thuê/ngày, Trạng thái (Sẵn sàng / Đang thuê / Bảo trì).
- Tạo danh sách quản lý các xe hiện có.
- Các thao tác: Thêm xe mới, Xóa xe, Cập nhật giá thuê, Tìm kiếm xe theo Biển số hoặc tìm các xe đang "Sẵn sàng".

Quản lý Khách hàng (1đ):

- Thông tin khách hàng: Mã khách hàng (CCCD/Passport), Họ tên, Số điện thoại, Năm sinh.
- Các thao tác: Thêm khách hàng mới, Cập nhật thông tin khách, Tìm kiếm khách hàng.

Quản lý Thuê & Trả xe (3đ):

- Thông tin Hợp đồng thuê gồm: Mã hợp đồng, Mã khách hàng, Biển số xe, Ngày thuê, Ngày trả dự kiến.
- Quy trình Thuê xe:
 - Kiểm tra trạng thái xe. Chỉ cho phép thuê nếu xe đang ở trạng thái "Sẵn sàng".
 - Sau khi tạo hợp đồng thành công, hệ thống tự động cập nhật trạng thái xe sang "Đang thuê".
- Quy trình Trả xe:
 - Nhập Biển số xe hoặc Mã hợp đồng để tìm thông tin.
 - Tính toán tổng tiền thuê = (Số ngày thuê thực tế) * (Giá thuê/ngày).
 - Cập nhật trạng thái xe về "Sẵn sàng" (hoặc "Bảo trì" nếu xe có vấn đề).
 - Lưu thông tin giao dịch vào lịch sử doanh thu.

Thống kê & Lịch sử (1.5đ):

- Tạo danh sách lưu trữ các hợp đồng đã hoàn tất (Lịch sử thuê).
- Thông tin lưu: Mã khách, Biển số xe, Ngày thuê, Ngày trả, Tổng tiền thanh toán.
- Chức năng: Tra cứu lịch sử thuê xe của một khách hàng cụ thể (để xem khách này đã thuê những xe gì, tổng tiền đã chi).

Yêu cầu các chức năng mở rộng:

Lưu trữ dữ liệu (1đ):

- Sử dụng file text để lưu trữ dữ liệu bền vững (bản chất giống như dùng file thay thế cho database), thay vì chỉ lưu vào vector mỗi khi khởi động chương trình là mất dữ liệu lần chạy trước -> không được tính điểm phần này.
- Khi chương trình bắt đầu (Start), đọc dữ liệu từ file vào các vector/List. Khi chương trình kết thúc (Exit) hoặc sau mỗi giao dịch quan trọng, ghi đè lại dữ liệu từ vector xuống file để đảm bảo không mất dữ liệu.

Quản lý ngoại lệ (Exception Handling) (1đ):

- Xây dựng các lớp ngoại lệ (custom exception) kế thừa từ std:: runtime_error:
 - CarNotAvailableException: Ném ra khi cố thuê một chiếc xe đang bận hoặc đang bảo trì.
 - InvalidIdException: Khi số CCCD hoặc Biển số xe nhập vào rỗng hoặc sai định dạng.
- Sử dụng khối try-catch tại hàm main hoặc menu để bắt lỗi và hiển thị thông báo thân thiện (không để chương trình bị crash).

Template Class (Generic Programming) (1đ):

- Xây dựng một lớp Template tên là QuanLy<T> (Manager).
- Lớp này sẽ chứa một std::vector<T> (hoặc cấu trúc danh sách tự định nghĩa) và chịu trách nhiệm cho các thao tác chung:
 - Them(T item): Thêm đối tượng vào danh sách.
 - Xoa(string id): Xóa đối tượng theo ID.
 - TimKiem(string id): Trả về đối tượng hoặc chỉ số.
 - HienThiDanhSach(): In toàn bộ danh sách.
- Áp dụng lớp QuanLy<T> này để quản lý: QuanLy<TaiKhoan>, QuanLy<Oto>, QuanLy<KhachHang>, QuanLy<HopDong>.

3. Đề tài 3: Xây dựng chương trình quản lý Cửa hàng Laptop/Điện thoại

Bạn hãy sử dụng các kiến thức về lập trình hướng đối tượng C++ để xây dựng chương trình hỗ trợ nhân viên bán hàng quản lý kho máy và lịch sử giao dịch.

Yêu cầu chức năng cơ bản:

Hệ thống Menu và Đăng nhập (0.5đ):

- Tạo danh sách tài khoản nhân viên bán hàng (Username, Password).
- Yêu cầu đăng nhập hợp lệ trước khi truy cập vào menu chức năng chính.

Quản lý Sản phẩm (1đ):

- Thông tin mỗi máy gồm: Mã Serial/IMEI (Mã duy nhất, không trùng lặp), Tên máy (VD: MacBook Pro M1, iPhone 14), Hãng sản xuất, Giá bán, Trạng thái (Còn hàng / Đã bán).
- *Lưu ý: Khác với tạp hóa (bán 10 gói mì, bán 2 đôi giày), cửa hàng điện tử quản lý từng chiếc máy cụ thể.*
- Các thao tác: Nhập kho máy mới (Thêm), Xóa máy (nếu nhập sai), Cập nhật giá bán, Tìm kiếm máy theo Serial hoặc Tên, Liệt kê các máy đang "Còn hàng".

Quản lý Khách hàng (1đ):

- Thông tin khách hàng: Mã khách hàng (CCCD/Passport), Họ tên, Số điện thoại, Năm sinh.
- Các thao tác: Thêm khách hàng mới, Cập nhật thông tin khách, Tìm kiếm khách hàng.

Quản lý Bán hàng (3đ):

- Thông tin Hóa đơn bán hàng: Mã hóa đơn, Số điện thoại khách, Mã Serial máy, Ngày bán, Giá thực tế (tức là giá lúc bán, có thể giảm giá so với giá niêm yết hiện tại đã thay đổi chẳng hạn).
- Quy trình Bán hàng:
 - Nhân viên nhập Mã Serial của máy cần bán.
 - Hệ thống kiểm tra: Chỉ cho phép bán nếu máy đang ở trạng thái "Còn hàng".
 - Nhập thông tin khách hàng (hoặc chọn khách cũ).
 - Sau khi tạo hóa đơn thành công, hệ thống tự động cập nhật trạng thái máy sang "Đã bán".
- Quy trình Tra cứu:
 - Nhập mã Serial để kiểm tra xem máy này đã bán chưa, bán ngày nào, cho ai.

Thống kê & Doanh thu (1.5đ):

- Tạo danh sách lưu trữ các Hóa đơn đã xuất (hóa đơn bao gồm những thông tin như ở phần trên).
- Chức năng:

- Tính tổng doanh thu của cửa hàng.
- Tra cứu lịch sử mua hàng của một khách hàng dựa trên Số điện thoại (Khách này đã mua những máy nào).

Yêu cầu các chức năng mở rộng:

Lưu trữ dữ liệu (1đ):

- Sử dụng file text để lưu trữ dữ liệu bền vững (bản chất giống như dùng file thay thế cho database), thay vì chỉ lưu vào vector mỗi khi khởi động chương trình là mất dữ liệu lần chạy trước -> không được tính điểm phần này.
- Khi chương trình bắt đầu (Start), đọc dữ liệu từ file vào các vector/List. Khi chương trình kết thúc (Exit) hoặc sau mỗi giao dịch quan trọng, ghi đè lại dữ liệu từ vector xuống file để đảm bảo không mất dữ liệu.

Quản lý ngoại lệ (Exception Handling) (1đ):

- Xây dựng các lớp ngoại lệ (custom exception) kế thừa từ `std::runtime_error`:
 - `ProductNotFoundException`: Khi nhập mã Serial không tồn tại trong kho.
 - `ProductAlreadySoldException`: Khi cố gắng bán một chiếc máy có trạng thái là "Đã bán".
 - `InvalidInfoException`: Khi để trống tên máy hoặc nhập giá tiền âm.
- Sử dụng khối try-catch tại hàm main hoặc menu để bắt lỗi và hiển thị thông báo thân thiện (không để chương trình bị crash).

Template Class (Generic Programming) (1đ):

- Xây dựng một lớp Template tên là `QuanLy<T>` (Manager).
- Lớp này sẽ chứa một `std::vector<T>` (hoặc cấu trúc danh sách tự định nghĩa) và chịu trách nhiệm cho các thao tác chung:
 - `Them(T item)`: Thêm đối tượng vào danh sách.
 - `Xoa(string id)`: Xóa đối tượng theo ID.
 - `TimKiem(string id)`: Trả về đối tượng hoặc chỉ số.
 - `HienThiDanhSach()`: In toàn bộ danh sách.

Áp dụng lớp `QuanLy<T>` này để quản lý: `QuanLy<TaiKhoan>`, `QuanLy<SanPham>`, `QuanLy<KhachHang>`, `QuanLy<HoaDon>`.