

W1: OOP

Câu 1

Xây dựng một hệ thống quản lý thư viện đơn giản với các yêu cầu sau:

```
def display_library_menu():
    print("\n=== LIBRARY MANAGEMENT SYSTEM ===")
    print("1. Add new book")
    print("2. Remove book")
    print("3. Borrow book")
    print("4. Return book")
    print("5. Display all books")
    print("6. Search book by ISBN")
    print("7. Exit")
    return input("Enter your choice (1-7): ")
```

- Tạo class Book với các thuộc tính: title, author, isbn, status (available/borrowed)
- Tạo class Library để quản lý danh sách sách với các phương thức:
 - add_book(): Thêm sách mới
 - remove_book(): Xóa sách theo ISBN
 - borrow_book(): Mượn sách
 - return_book(): Trả sách
 - display_books(): Hiện thị danh sách sách

Câu 2

Xây dựng hệ thống tài khoản ngân hàng với các yêu cầu:

```
def display_banking_menu():
    print("\n=== BANKING SYSTEM ===")
    print("1. Create new account")
    print("2. Display account info")
    print("3. Deposit")
    print("4. Withdraw")
    print("5. Check balance")
    print("6. Transfer money")
    print("7. List all accounts")
    print("8. Exit")
    return input("Enter your choice (1-8): ")
```

- Tạo class Account với các thuộc tính: account_number, owner_name, balance
- Tạo các phương thức:
 - deposit(): Nạp tiền
 - withdraw(): Rút tiền (có kiểm tra số dư)
 - get_balance(): Xem số dư
 - transfer(): Chuyển tiền sang tài khoản khác
- Áp dụng tính đóng gói để bảo vệ các thuộc tính
- Tạo phương thức để in thông tin tài khoản

Câu 3

Xây dựng hệ thống quản lý nhân viên với các yêu cầu:

```
def display_employee_menu():
    print("\n=== EMPLOYEE MANAGEMENT SYSTEM ===")
    print("1. Add new employee")
    print("2. Remove employee")
    print("3. Display all employees")
    print("4. Calculate payroll")
    print("5. Search employee by ID")
    print("6. Display employees by department")
    print("7. Update employee information")
    print("8. Exit")
    return input("Enter your choice (1-8): ")
```

- Tạo class abstract Employee với các thuộc tính cơ bản: id, name, salary
- Tạo các class con: Manager, Developer, Tester kế thừa từ Employee
- Mỗi loại nhân viên có cách tính lương riêng:
 - Manager: lương cơ bản + phụ cấp quản lý
 - Developer: lương cơ bản + thưởng dự án
 - Tester: lương cơ bản + thưởng bug
- Tạo class Department để quản lý danh sách nhân viên
- Thêm các phương thức:
 - add_employee(): Thêm nhân viên mới
 - remove_employee(): Xóa nhân viên
 - calculate_payroll(): Tính tổng lương phải trả
 - display_employees(): Hiện thị danh sách nhân viên

Yêu cầu chung:

- ☐ Áp dụng đúng các nguyên tắc OOP:
 - Sử dụng tính kế thừa để tái sử dụng code
 - Áp dụng tính đa hình qua các abstract methods
 - Đảm bảo tính đóng gói bằng private/protected attributes
 - Thiết kế các abstract classes hợp lý
- ☐ Xử lý các trường hợp ngoại lệ
- ☐ Đặt tên biến, phương thức rõ ràng, dễ hiểu