

Phần I

LÝ THUYẾT

1. Cơ sở lý thuyết

- 1.1. Lập trình hướng đối tượng có mấy đặc trưng (mỗi đặc trưng cho một ví dụ)
- 1.2. Lớp là gì? Cho ví dụ
- 1.3. Nêu các phương thức có thể có trong một lớp
- 1.4. Cú pháp của phương thức
- 1.5. Hàm bạn là gì? Cú pháp khai báo hàm b
- 1.6. Getter, setter là gì? Cho ví dụ
- 1.7. Nạp chồng toán tử là gì? Vì sao phải nạp chồng
- 1.8. Kế thừa là gì có mấy loại kế thừa, mục đích của kế thừa. cho ví dụ
- 1.9. Đa hình là gì. Cho ví dụ
- 1.10. Interface là gì? Cho ví dụ

Phần 2: bài tập

Câu 1: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **thí sinh** với các thuộc tính: mã thí sinh, họ tên, điểm toán, điểm lý, điểm hóa.
2. Xây dựng các phương thức: nhập, xuất, tính tổng điểm một đối tượng thí sinh.
Xây dựng constructor không đối, có đối
3. Nhập vào n thí sinh. In ra màn hình thông tin của thí sinh có tổng điểm cao nhất (yêu cầu sử dụng biến con trỏ để cấp phát và giải phóng bộ nhớ).
4. Sắp xếp danh sách theo họ tên tăng dần
5. Tìm kiếm sinh viên theo họ tên

Câu 2: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **đa thức** với các thuộc tính: bậc đa thức, các hệ số tương ứng.
2. Xây dựng các phương thức: nhập, xuất một đối tượng đa thức.
3. Định nghĩa toán tử +, - hai đa thức. Thực hiện cộng, trừ hai đa thức và in kết quả ra màn hình.

Câu 3: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **date** với các thuộc tính: ngày, tháng, năm.

Xây dựng hàm tạo không đối, hàm tạo có đối, hàm hủy một đối tượng date.

2. Xây dựng các phương thức: nhập, xuất, chuẩn hóa một đối tượng date.

Định nghĩa toán tử ++,-- để tăng, giảm một ngày. Thực hiện tăng, giảm một ngày và in kết quả ra màn hình (kết quả phải được chuẩn hóa dưới dạng dd-mm-yyyy).

Int a;

A++;//thực hiện biểu thức A trước sau đó mới tăng a lên một đơn vị

++A; // Tăng A trước lên 1 một đơn sau rồi mới thực hiện biểu thức

Câu 4:

3. Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **time** với các thuộc tính: giờ, phút, giây.

Xây dựng hàm tạo không đối, hàm tạo có đối, hàm hủy một đối tượng time.

2. Xây dựng các phương thức: nhập, xuất, chuẩn hóa một đối tượng time.

3. Định nghĩa toán tử ++,-- để tăng, giảm một giây. Thực hiện tăng, giảm một giây và in kết quả ra màn hình (kết quả phải được chuẩn hóa dưới định dạng 24 tiếng hh:mm:ss).

Câu 5: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **người** với các thuộc tính: họ tên, năm sinh.

Khai báo lớp **sinh viên** kế thừa từ lớp người và thêm các thuộc tính: mã sinh viên, điểm trung bình.

2. Xây dựng các phương thức: nhập, xuất cho các đối tượng người, sinh viên.
3. Nhập vào n sinh viên. Sắp xếp lại và in ra màn hình danh sách sinh viên theo thứ tự giảm dần của điểm trung bình.

Câu 6: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **người** với các thuộc tính: họ tên, năm sinh.

Khai báo lớp **sinh viên** kế thừa từ lớp người và thêm các thuộc tính: mã sinh viên, điểm trung bình.

2. Xây dựng các phương thức: constructor, nhập, xuất cho các đối tượng người, sinh viên.
3. Nhập vào n sinh viên. Nhập thông tin tìm kiếm là mã sinh viên hoặc họ tên. In ra màn hình thông tin sinh viên tìm thấy hoặc thông báo là không tìm thấy.

Câu 7: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **ma trận vuông** với các thuộc tính: cấp ma trận, các phần tử của ma trận.
2. Xây dựng các phương thức: nhập, xuất một đối tượng ma trận.
3. Nhập vào 1 ma trận. In ra màn hình ma trận vừa nhập. Tính tổng các phần tử nằm trên đường chéo chính, tổng các phần tử nằm trên đường chéo phụ của ma trận.

Câu 8: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **ma trận** với các thuộc tính: số hàng, số cột, các phần tử của ma trận.
2. Xây dựng các phương thức: nhập, xuất một đối tượng ma trận.
3. Nhập vào 1 ma trận. In ra màn hình ma trận vừa nhập. Tính tổng các phần tử nằm trên 4 đường viền của ma trận.

Câu 9: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **nhân viên** với các thuộc tính: họ tên, chức vụ, giới tính, năm sinh, quê quán, năm vào làm.
2. Xây dựng các toán tử >>, << để nhập, xuất một đối tượng nhân viên.
3. Nhập vào n nhân viên. In ra màn hình danh sách các nhân viên làm việc trên 20 năm (số năm làm việc bằng năm hiện tại trừ năm vào làm).

Câu 10: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **nhân sự** với các thuộc tính: mã nhân viên, họ tên, năm sinh.
Khai báo lớp **cán bộ** kế thừa từ lớp nhân sự và thêm các thuộc tính: lương cơ bản, hệ số lương.
2. Xây dựng các phương thức: nhập, xuất cho các đối tượng nhân sự, cán bộ.
3. Nhập vào n cán bộ. Thực hiện tính lương cho các cán bộ. In ra màn hình danh sách cán bộ theo thứ tự giảm dần của tiền lương.

Câu 11: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **nhân sự** với các thuộc tính: mã nhân viên, họ tên, năm sinh.
Khai báo lớp **công nhân** kế thừa từ lớp nhân sự và thêm thuộc tính: ngày công.
2. Xây dựng các phương thức: nhập, xuất cho các đối tượng nhân sự, công nhân.

Nhập vào n công nhân. Thực hiện tính tiền thưởng cho công nhân 100.000 với mỗi ngày công làm thêm nếu ngày công lớn hơn 26. In ra màn hình danh sách các công nhân được thưởng và số tiền thưởng tương ứng.

Câu 12: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **môn học** với các thuộc tính: tên môn học, điểm chuyên cần (CC), điểm kiểm tra (KT), điểm thi (DT).
Khai báo lớp **sinh viên** kế thừa từ lớp môn học và thêm các thuộc tính: họ tên, lớp, mã sinh viên.
2. Xây dựng các phương thức: nhập, xuất, tính điểm học phần cho một sinh viên.
3. Nhập vào n sinh viên. In ra màn hình thông tin các sinh viên bị cấm thi (điểm chuyên cần dưới 5 hoặc điểm kiểm tra bằng 0).

Câu 13: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **ma trận** với các thuộc tính: số hàng, số cột, các phần tử của ma trận.
2. Xây dựng các phương thức: nhập, xuất một đối tượng ma trận.
3. Nhập vào hai ma trận đồng cấp. Thực hiện cộng hai ma trận vừa nhập (dùng hàm bạn). In ra màn hình hai ma trận ban đầu và ma trận kết quả.

Câu 14: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **người** với các thuộc tính: họ tên, năm sinh.
Khai báo lớp **giảng viên** kế thừa từ lớp người và thêm các thuộc tính: bộ môn công tác, môn đang giảng dạy.
2. Xây dựng các phương thức: nhập, xuất cho các đối tượng người, giảng viên.
3. Nhập vào n giảng viên. Sắp xếp lại và in ra màn hình danh sách giảng viên theo từng bộ môn công tác.

Câu 15: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **cán bộ** với các thuộc tính: mã cán bộ, mã đơn vị, họ tên, năm sinh.
Khai báo lớp **lương** kế thừa từ lớp cán bộ và thêm các thuộc tính: phụ cấp, hệ số lương, bảo hiểm.
2. Xây dựng các phương thức: nhập, xuất cho các đối tượng cán bộ, lương.

Nhập vào n cán bộ. Tính lương cho cán bộ theo công thức: $\text{lương} = \text{hệ số lương} * 290.000 + \text{phụ cấp} - \text{bảo hiểm}$. In kết quả ra màn hình

Câu 16: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **hình vuông** với thuộc tính: độ dài cạnh.
Khai báo lớp **hình chữ nhật** kế thừa từ lớp hình vuông và thêm thuộc tính: độ dài cạnh thứ 2.
2. Xây dựng các phương thức: nhập, xuất, tính chu vi, diện tích hình vuông, hình chữ nhật.
3. Nhập vào n hình vuông. In ra màn hình hình vuông có diện tích lớn nhất và vị trí của hình vuông đó trong danh sách vừa nhập.

Câu 17: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **điểm** với các thuộc tính: tọa độ x, tọa độ y.
Khai báo lớp **tam giác** có thuộc tính 3 đỉnh là kiểu đối tượng điểm (lớp tam giác được gọi là lớp bao của lớp điểm) và thêm các thuộc tính: độ dài 3 cạnh.
2. Xây dựng các phương thức: nhập, xuất, tính khoảng cách 2 điểm (dùng hàm bạn), áp dụng tính độ dài 3 cạnh của tam giác; nhập, xuất 3 đỉnh của tam giác.
3. Nhập vào 1 tam giác. Kiểm tra tính chất của tam giác (thường, đều, vuông, cân, vuông cân). In kết quả ra màn hình.

Câu 18: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **PS1** cho các đối tượng là phân số với các thuộc tính: tử số, mẫu số.
Xây dựng phương thức nhập phân số (mẫu số khác 0), in phân số, tối giản phân số.
2. Xây dựng lớp **PS2** kế thừa từ lớp **PS1** và bổ sung:
Nạp chồng các toán tử: $=$ (gán), $>$ (lớn hơn).
3. Viết chương trình chính ứng dụng lớp **PS2** để nhập một danh sách các đối tượng là phân số (tối đa 10 phần tử). Sắp xếp lại danh sách đã nhập theo trật tự giảm dần.

Câu 19: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **PS1** cho các đối tượng là phân số với các thuộc tính: tử số, mẫu số.
Xây dựng phương thức nhập phân số (mẫu số khác 0), in phân số, tối giản phân số.
2. Xây dựng lớp **PS2** kế thừa từ lớp **PS1** và bổ sung:
Nạp chồng các toán tử: $=$ (gán), $<$ (nhỏ hơn).
3. Viết chương trình chính ứng dụng lớp **PS2** để nhập một danh sách các đối tượng là các phân số (tối đa 10 phần tử). Tìm phân số có giá trị lớn nhất, nhỏ nhất.

Câu 20: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **SP1** cho các đối tượng là số phức với các thuộc tính: phần thực, phần ảo.
Xây dựng hàm tạo, phương thức nhập số phức, in số phức, tính module số phức.
2. Xây dựng lớp **SP2** kế thừa từ lớp **SP1** và bổ sung:
Nạp chồng các toán tử: $=$ (gán), $>$ (so sánh lớn hơn theo module).
3. Viết chương trình chính ứng dụng lớp **SP2** để nhập một danh sách các đối tượng là các số phức (tối đa 10 phần tử). Sắp xếp lại danh sách đã nhập theo trật tự giảm dần của module.

Câu 21: Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **SP1** cho các đối tượng là số phức với các thuộc tính: phần thực, phần ảo.

Xây dựng hàm tạo, phương thức nhập số phức, in số phức, tính module số phức.

2. Xây dựng lớp **SP2** kế thừa từ lớp **SP1** và bổ sung:
Nạp chồng các toán tử: $=$ (gán), $<$ (so sánh nhỏ hơn theo module).
3. Viết chương trình chính ứng dụng lớp **SP2** để nhập một danh sách các đối tượng là các số phức (tối đa 10 phần tử). Sắp xếp lại danh sách đã nhập theo trật tự tăng dần của module.

Câu 22:

Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **SP1** cho các đối tượng là số phức với các thuộc tính: phần thực, phần ảo.
Xây dựng hàm tạo, phương thức nhập số phức, in số phức, tính module số phức.
2. Xây dựng lớp **SP2** kế thừa từ lớp **SP1** và bổ sung:
Nạp chồng các toán tử = (gán), < (so sánh nhỏ hơn theo module)
3. Viết chương trình chính ứng dụng lớp **SP2** để nhập một danh sách các đối tượng là các số phức (tối đa 10 phần tử). Tìm số phức có giá trị lớn nhất, nhỏ nhất.

Câu 23:

Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **PS1** cho các đối tượng là phân số với các thuộc tính: tử số, mẫu số.
Xây dựng phương thức nhập phân số (mẫu số khác 0), in phân số, tối giản phân số.
2. Xây dựng lớp **PS2** kế thừa từ lớp **PS1** và bổ sung:
Nạp chồng các toán tử: = (gán), < (nhỏ hơn), + (cộng).
3. Viết chương trình chính ứng dụng lớp **PS2** để nhập một danh sách các đối tượng là các phân số (tối đa 10 phần tử). Tìm phân số có giá trị lớn nhất, tính tổng các phân số trong danh sách có giá trị nhỏ hơn $1/2$.

Câu 24:

Viết chương trình thực hiện các yêu cầu sau:

1. Khai báo lớp **PS1** cho các đối tượng là phân số với các thuộc tính: tử số, mẫu số.
Xây dựng phương thức nhập phân số (mẫu số khác 0), in phân số, tối giản phân số.
2. Xây dựng lớp **PS2** kế thừa từ lớp **PS1** và bổ sung:
Nạp chồng các toán tử: = (gán), > (lớn hơn), + (cộng).

- Viết chương trình chính ứng dụng lớp **PS2** để nhập một danh sách các đối tượng là các phân số (tối đa 10 phân tử). Tìm phân số có giá trị lớn nhất, tính tổng các phân số trong danh sách có giá trị lớn hơn $1/4$.

Câu 25: Viết chương trình thực hiện các yêu cầu sau:

- Khai báo lớp **SP1** mô tả các đối tượng là số phức với các thuộc tính: phần thực, phần ảo.
Xây dựng hàm tạo, phương thức nhập số phức, in số phức, tính module số phức
- Xây dựng lớp **SP2** kế thừa từ lớp **SP1** và bổ sung:
Nạp chồng các toán tử: = (gán), < (nhỏ hơn), + (cộng)
- Viết chương trình chính ứng dụng lớp **SP2** để nhập một danh sách các đối tượng là các số phức (tối đa 10 phân tử). Tìm số phức có giá trị nhỏ nhất (theo module) và tính tổng các số phức trong dãy số.

Câu 26: Viết chương trình thực hiện các yêu cầu sau:

- Khai báo lớp **SP1** mô tả các đối tượng là số phức với các thuộc tính: phần thực, phần ảo.
Xây dựng hàm tạo, phương thức nhập số phức, in số phức, tính module số phức
- Xây dựng lớp **SP2** kế thừa từ lớp **SP1** và bổ sung:
Nạp chồng các toán tử: = (gán), > (nhỏ hơn), == (so sánh bằng)
- Viết chương trình chính ứng dụng lớp **SP2** để nhập một danh sách các đối tượng là các số phức (tối đa 10 phân tử). Tìm số phức có giá trị lớn nhất và đếm xem có bao nhiêu số phức trong danh sách có giá trị bằng $3+4i$.

Số phức : $a+bi$: trong a số thực b là số ảo

Module $z=\sqrt{a^2+b^2}$

$Z1=a1+b1*i$

$Z2=a2+b2*i$

$Z1=Z2$: $a1==a2$ và $b1==b2$

$Z1>Z2$: khi module $z1>module\ z2$

Câu 27: Nhân viên trong một cơ quan được lĩnh lương theo các dạng khác nhau. Dạng người lao động hưởng lương từ ngân sách Nhà nước gọi là cán bộ, công chức (dạng biên chế). Dạng người lao động lĩnh lương từ ngân sách của cơ quan gọi là người làm hợp đồng. Như vậy, trong hệ thống cấp phát lương có 2 dạng: biên chế và hợp đồng. Hai loại đối tượng này có đặc tính chung là viên chức làm việc cho cơ quan. Trong bảng lương với mỗi viên chức, cần quản lý các thông tin: Mã số, họ tên, chức vụ. Ngoài ra với viên chức là biên chế cần biết thêm thông tin: Hệ số lương, với mỗi viên chức là hợp đồng cần biết thêm thông tin: số ngày làm việc trong tháng.

Anh (Chị) hãy viết chương trình hướng đối tượng, trong đó xây dựng lớp cơ sở và các lớp dẫn xuất với các dữ liệu thành viên và các hàm cần thiết để thực hiện các yêu cầu sau:

1. Nhập dữ liệu cho cán bộ, nhân viên trong biên chế và nhân viên hợp đồng. Dữ liệu được lưu trong mảng các đối tượng.
2. In ra màn hình bảng lương theo mẫu:

BẢNG DANH SÁCH TIỀN LƯƠNG CÁN BỘ BIÊN CHẾ

STT	Họ tên	Hệ số lương	Tiền lương
-----	--------	-------------	------------

BẢNG DANH SÁCH TIỀN LƯƠNG CÁN BỘ HỢP ĐỒNG

STT	Họ tên	Số ngày làm việc	Tiền lương
-----	--------	------------------	------------

Biết rằng: Tiền lương (cán bộ biên chế) = Hệ số lương x 830000 + PCCV - BHXH

Tiền lương (cán bộ hợp đồng) = 1500000 + tiền tăng giờ - BHXH

BHXH = 5% * Tiền lương

PCCV (phụ cấp chức vụ) được tính theo quy định: Giám đốc: 150000, Phó giám đốc: 100000, các chức vụ khác không có PCCV.

Tiền tăng giờ: Nếu cán bộ hợp đồng làm việc trên 26 ngày /tháng thì mỗi ngày dôi ra được tính bồi dưỡng 100000 đ/ngày, làm việc dưới 26 ngày/tháng thì mỗi ngày nghỉ bị trừ 50000đ.

Câu 28: Khởi động C++, tạo tập tin có tên D:\Bai1.CPP, sau đó viết chương trình quản lý thi chứng chỉ tin học, theo yêu cầu:

- Chương trình có 3 lớp:
 - + Lớp cơ sở **Hoso** lưu Ma (mã học sinh), Hodem (họ đệm), Ten (tên).
 - + Lớp dẫn xuất **Diem** kế thừa lớp **Hoso** lưu dlt (điểm lý thuyết), dth (điểm thực hành).
 - + Lớp dẫn xuất **Ketqua** kế thừa lớp **Diem**.
- Trong lớp **Hoso** có các hàm thành viên: **Nhap_HS()** và **Hienthi_HS()** để nhập và hiển thị các dữ liệu thành viên của lớp **Hoso**.
- Trong lớp **Diem** có các hàm thành viên: **Nhap_d()** để nhập dữ liệu thành viên của lớp **Diem** và hàm **Hienthi_d()** để hiển thị dữ liệu thành viên của lớp **Diem()**.
- Trong lớp **Ketqua** chỉ có hàm **Hienthi_KQ()** cho phép hiển thị danh sách thí sinh đậu (các thí sinh có điểm lý thuyết ≥ 5 và điểm thực hành ≥ 5)

Trong hàm main() khởi tạo các đối tượng cần thiết và tiến hành cho nhập thông tin của n thí sinh từ bàn phím. Sau đó, chương trình cho phép hiển thị lên màn hình danh sách thí sinh đậu kỳ thi chứng chỉ.

Câu 29: Khởi động C++, tạo tập tin có tên D:\Bai1.CPP, sau đó viết chương trình thực hiện yêu cầu: Xây dựng lớp ps (phân số) có các thành phần:

- Các thuộc tính: tu, mau.
- Hàm tạo để thiết lập giá trị ban đầu cho các thuộc tính.
- Hàm nhap để nhập dữ liệu vào cho 2 dữ liệu thuộc tính.
- Hàm hienthi để in kết quả của 2 thuộc tính lên màn hình.
- Quá tải các toán tử $==$, $>$ để so sánh 2 phân số, $+$ để cộng hai phân số.

Sau đó xử lý thực hiện các công việc :

- a) Nhập n phân số từ bàn phím.
- b) In ra màn hình n phân số vừa nhập.
- c) In ra màn hình tổng của n phân số đã nhập (sau khi rút gọn phân số tổng).
- d) In ra màn hình phân số có giá trị lớn nhất.

e) Nhập một phân số từ bàn phím và tìm xem phân số đó đã có trong n phân số đã nhập ở trên không ?

Câu 30: Khởi động C++, tạo tập tin có tên Bai1.CPP, sau đó viết chương trình hướng đối tượng để đổi số nguyên dương ở hệ thập phân sang hệ cơ số bất kỳ (2, 8, 16) theo yêu cầu chương trình có 2 lớp:

+ Lớp cơ sở **Stack** có các dữ liệu thành viên và các hàm cần thiết: kiểm tra stack đầy, kiểm tra stack rỗng, nộp một phần tử vào stack, lấy một phần tử ra khỏi stack.

+ Lớp dẫn xuất **Doicoso** kế thừa lớp cơ sở **Stack** có hàm thành viên: **Doi** để đổi số nguyên dương ở hệ thập phân sang hệ cơ số bất kỳ, hàm **Hienthi()** để hiện thị số đã đổi ra màn hình.

Trong hàm main() khởi tạo các đối tượng cần thiết và tiến hành cho nhập thông tin của số nguyên dương n và cơ số muốn đổi từ bàn phím. Sau đó, chương trình gọi các hàm trong các lớp đã xây dựng, thực hiện chuyển đổi và in kết quả ra màn hình.

