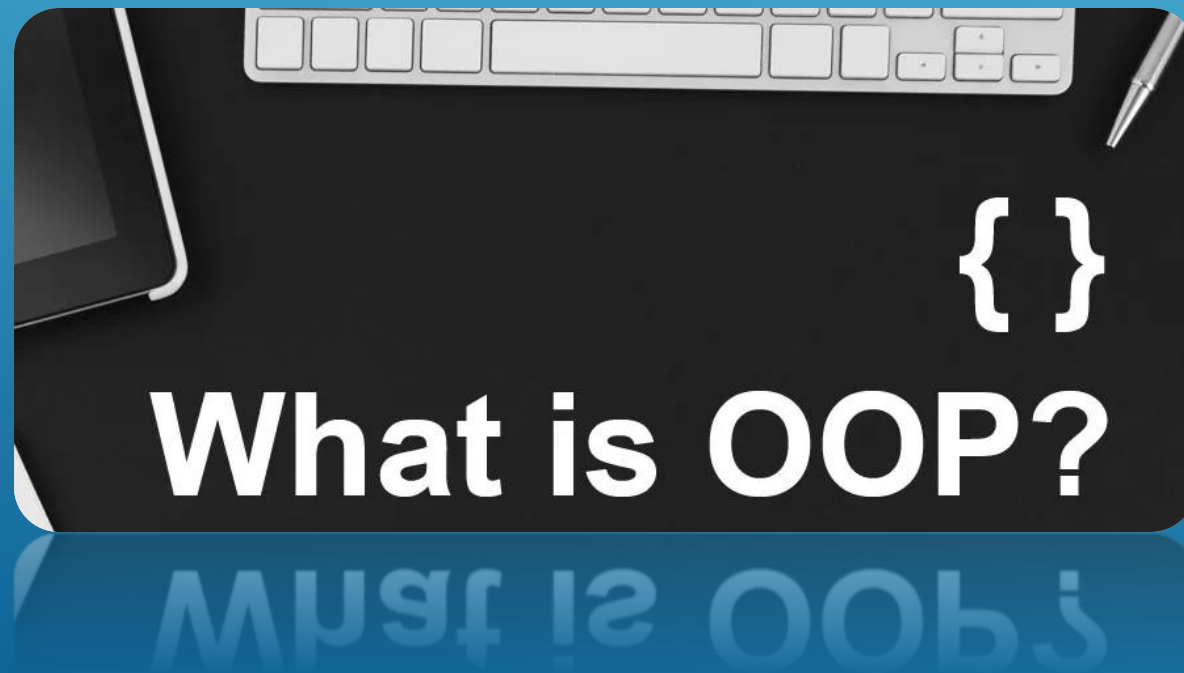


# BÀI BÁO CÁO

*Đề tài: Xây dựng chương trình  
quản lý sinh viên*

Quản lý sinh viên luôn luôn là một công việc hàng đầu của các trường đại học. Với lượng sinh viên hàng năm nhập học cũng như ra trường đông đảo thì việc quản lý những thông tin cũng như điểm của sinh viên rất quan trọng. Cùng với sự phát triển của công nghệ nói chung và Công nghệ thông tin nói riêng thì việc quản lý sinh viên cũng ngày càng được hiện đại hoá. Thay vì chúng ta phải ghi bằng chữ bằng những giấy tờ thì giờ đây chúng ta có thể sử dụng những phần mềm để giúp việc quản lý sinh viên tiện hơn, nhanh chóng hơn, tiết kiệm được thời gian hơn. Phần mềm quản lý sinh viên là phần mềm được tạo ra với mục tiêu là giúp đỡ giảng viên thuận tiện hơn trong việc quản lý điểm sinh viên cụ thể như xem, sửa, thêm hoặc xóa thông tin của sinh viên. Giảng viên chỉ cần đăng nhập vào phần mềm với một tài khoản được cấp, ngay lập tức với những cú click chuột đơn giản giờ đây giảng viên đã có thể truy cập vào hồ sơ của sinh viên một cách nhanh chóng và tiết kiệm thời gian. Chính vì vậy, đây là lý do em chọn đề tài quản lý sinh viên.



# lập trình hướng đối tượng

4 khái niệm bạn phải biết



Trước hết, hãy cùng tìm hiểu OOP là gì? OOP là từ viết tắt cho cụm từ Object Oriented Programming. Object Oriented Programming nghĩa là lập trình hướng đối tượng. Đây là một cách lập trình máy tính hay còn gọi là thiết kế phần mềm, lập trình viên sẽ xác định dạng dữ liệu của một cấu trúc dữ liệu, loại hàm có thể áp dụng cho cấu trúc dữ liệu đó. Bằng cách này, cấu trúc dữ liệu trở thành một đối tượng bao gồm cả dữ liệu và hàm. Tính chất của OOP có 4 tính chất bao gồm:

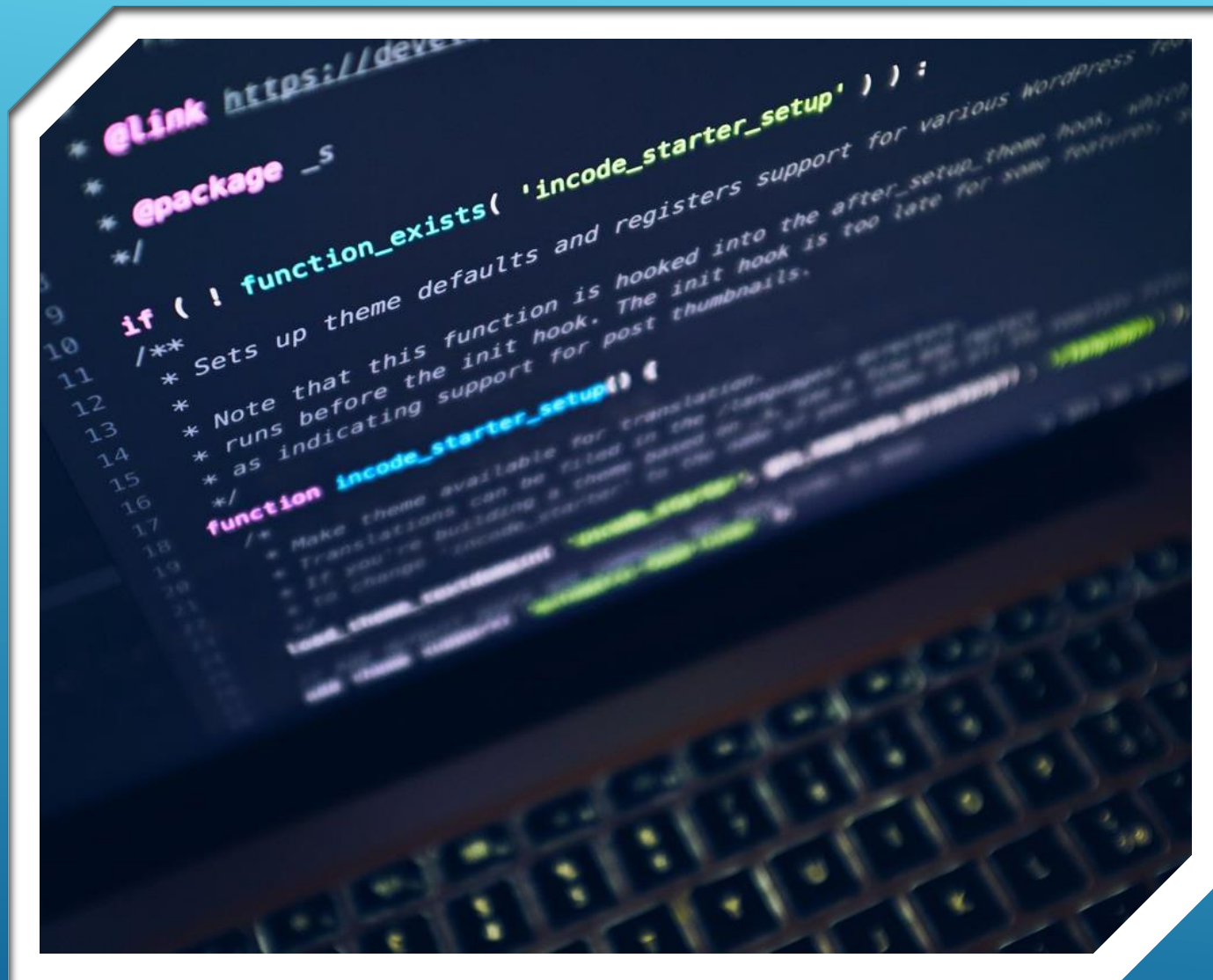
- ❑ *Tính đóng gói*
- ❑ *Tính trừu tượng*
- ❑ *Tính kế thừa*
- ❑ *Tính đa hình*

**Vì vậy, 4 tính chất trên là 4 trụ cột của OOP.**

## 1) TÍNH ĐÓNG GÓI

Các đối tượng khác nhau của mỗi chương trình sẽ cố gắng tương tác với nhau một cách tự động. Nếu một lập trình viên muốn ngăn cản sự tương tác giữa các đối tượng thì phải đóng gói các đối tượng vào những class riêng biệt. Thông qua đóng gói, các class sẽ không thay đổi và tương tác với những biến hay hàm cụ thể của một đối tượng.

```
9 ////////////////
10 class People {
11     protected:
12         char hoten[20], noisinh[70], gioitinh[10];
13         int dd, mm, yyyy;
14 };
15 ////////////////
16 class Diem{
17     protected:
18         string ten_mon_hoc[8];
19         float mon_hoc[8];
20         int so_tin_chi[8];
21         int size_mh;
22 };
23 ////////////////
24 class Sinhvien :public People,public Diem {
25     private:
26         int MSSV;
27         char nganh[30];
28         float dtb;
29     public:
30         Sinhvien(); //hàm tạo không tham số
31         Sinhvien(int, char [20], int, int, int, char [10] , char [70],
32         char [30], int, string [8], float [8], int [8], float ); //hàm tạo có tham số
33         Sinhvien(const Sinhvien&); //hàm tạo sao chép
34         ~Sinhvien(); //hàm hủy
35         int getMSSV(); //trả về MSSV
36         int getNgay(); //trả về ngày sinh
37         int getThang(); //trả về tháng sinh
38         float getDtb(); //trả về điểm trung bình
```



## 2) TÍNH TRỪU TƯỢNG

► Tính trừu tượng giống như một phiên bản mở rộng của tính đóng gói vì nó giấu đi những tính chất và phương thức cụ thể để giao thức của các đối tượng đơn giản hơn. Lập trình viên sử dụng tính trừu tượng cho vài lý do có ích khác. Nhìn chung, tính trừu tượng giúp cô lập ảnh hưởng của sự thay đổi mã code. Mục tiêu là nếu có sai sót gì xảy ra, ảnh hưởng của sự thay đổi là không nhiều.

► (Nguồn hình ảnh: Unsplash)



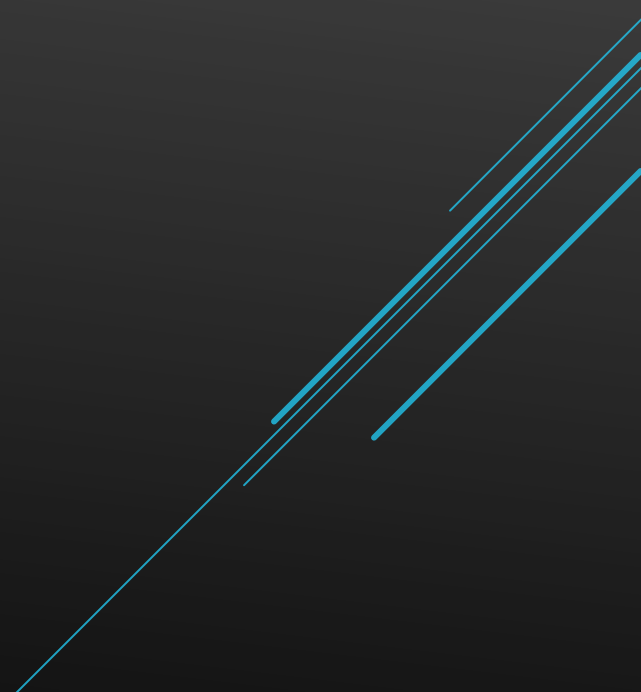
- ❑ Tính đóng gói và tính trừu tượng giúp chúng ta phát triển và duy trì một cơ sở mã lớn. Nhưng một vấn đề của OOP là các đối tượng thường rất giống nhau. Chúng có chung logic thực hiện, nhưng không thật sự giống nhau.
- ❑ Để tái sử dụng logic chung này và trích những logic cụ thể vào một class riêng, bạn có thể sử dụng tính kế thừa. Có nghĩa là bạn tạo ra một lớp con từ lớp cha. Bằng cách này, chúng ta tạo được một trật tự. Lớp con sẽ sử dụng tất cả các field và method của lớp cha và có thể có thêm những phần riêng biệt của nó.

### 3) TÍNH KẾ THỪA

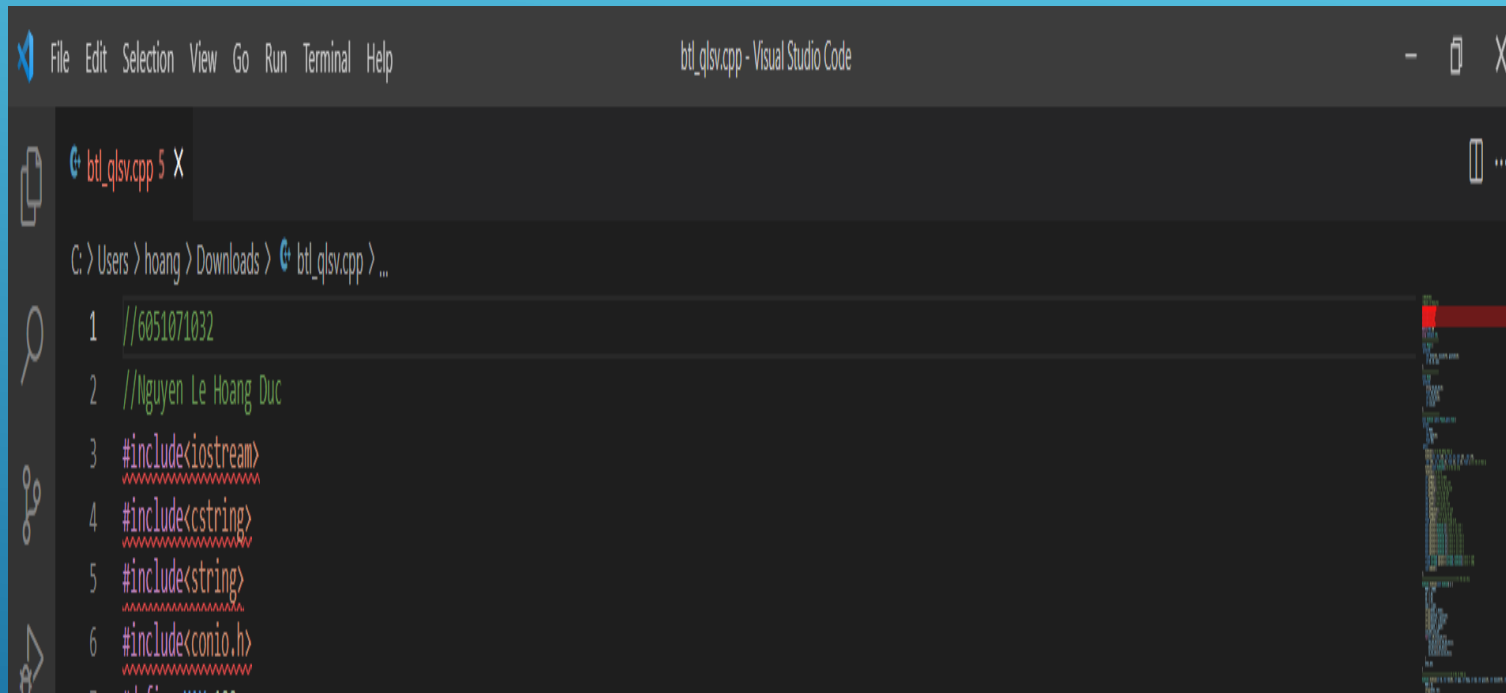
Động Vật	Con Trau
<b>Thuộc Tính:</b> <ul style="list-style-type: none"><li>- Mắt</li><li>- Mũi</li><li>- Miệng</li><li>- Chân</li><li>- Giới tính</li></ul>	<b>Thuộc Tính:</b> <ul style="list-style-type: none"><li>- Mắt</li><li>- Mũi</li><li>- Miệng</li><li>- Chân</li><li>- Giới Tính</li><li>- Sừng</li></ul>
<b>Phương Thức:</b> <ul style="list-style-type: none"><li>- Ăn</li><li>- Ngủ</li><li>- Chạy</li><li>- La hét</li></ul>	<b>Phương Thức:</b> <ul style="list-style-type: none"><li>- Ăn</li><li>- Ngủ</li><li>- Chạy</li><li>- La Hét</li></ul>

## 4) TÍNH ĐA HÌNH

Tính đa hình đưa ra cách sử dụng một lớp con giống hệt như lớp cha để không có sự nhầm lẫn, bởi rồi nào giữa các dạng khác nhau. Nhưng mỗi lớp con vẫn giữ nguyên method của mình. Điều này thường xảy ra khi tái sử dụng một giao thức lớp cha. Nó đưa ra những method phổ biến, rồi mỗi lớp con thực hiện phiên bản method riêng của nó.



# CHƯƠNG TRÌNH QUẢN LÝ SINH VIÊN



```
File Edit Selection View Go Run Terminal Help
btl_qslv.cpp - Visual Studio Code

btl_qslv.cpp
C:\Users\hoang\Downloads> btl_qslv.cpp > ...

1 //6051071032
2 //Nguyen Le Hoang Duc
3 #include<iostream>
4 #include<cstring>
5 #include<string>
6 #include<conio.h>
```

## 1. Các thư viện:

`#include<iostream>`

`#include<cstring>`

`#include<string>`

`#include<conio.h>`



```
class People {  
    protected:  
        char hoten[20], noisinh[70], gioitinh[10];  
        int dd, mm, yyyy;  
};
```

## 2) CÁC LỚP:

- ❖ Lớp **People**: gồm các thuộc tính tên (char hoten[20]), giới tính (char gioitinh[10]), nơi sinh (char noisinh[70]), ngày tháng năm sinh (int dd, mm, yyyy)

- ❖ Lớp **Diem**: gồm các thuộc tính tên môn học (char ten\_mon\_hoc[30][8]), điểm (float mon\_hoc[8]), số tín chỉ (int so\_tin\_chi[8]), số lượng môn học (int size\_mh).

```
class Diem {  
protected:  
    string ten_mon_hoc[8];  
    float mon_hoc[8];  
    int so_tin_chi[8];  
    int size_mh;  
};
```

- ❖ Lớp **Sinhvien**: kế thừa 2 lớp **People** và **Diem** và có các thuộc tính riêng gồm: mã số sinh viên (int MSSV), ngành học (char nganh[30]), điểm trung bình (float dtb);

```
class Sinhvien :public People,public Diem
private:
    int MSSV;
    char nganh[30];
    float dtb;
```

```
public:
    Sinhvien();//hàm tạo không tham số
    Sinhvien(int, char [20], int, int, int, char [10] , char [70],
    char [30], int, string [8], float [8], int [8], float );//hàm tạo có tham số
    Sinhvien(const Sinhvien&);//hàm tạo sao chép
    ~Sinhvien();//hàm hủy
    int getMSSV();//trả về MSSV
    int getNgay();//trả về ngày sinh
    int getThang();//trả về tháng sinh
    int getNam();//trả về năm sinh
    char getTen();//trả về họ tên
```

```
char getGT();//trả về giới tính
char getNoiSinh();//trả về nơi sinh
char getNganh();//trả về ngành học
float getdtb();//trả về điểm trung bình
bool operator>(Sinhvien sv);//toán tử so sánh >
bool operator<(Sinhvien sv);//toán tử so sánh <
bool operator==(Sinhvien sv);//toán tử so sánh ==
bool operator>=(Sinhvien sv);//toán tử so sánh >=
bool operator<=(Sinhvien sv);//toán tử so sánh <=
bool operator!=(Sinhvien sv);//toán tử so sánh !=
friend ostream& operator<<(ostream&, Sinhvien&);//toán tử xuất
friend istream& operator>>(istream&, Sinhvien&);//toán tử nhập
void xemdiem();
};
```

```
Sinhvien::Sinhvien() {  
    dd = yyyy = mm = 0;  
    strcpy(hoten, "");  
    strcpy(gioitinh, "");  
    strcpy(noisinh, "");  
    MSSV = 0;  
    strcpy(nganh, "");  
    size_mh=0;  
    dtb=0.0;  
}
```

### *3. CÁC PHƯƠNG THỨC LỚP SINH VIÊN:*

#### **A) HÀM TẠO:**

- ✓ Hàm tạo không tham số:

```
Sinhvien::Sinhvien(int Id, char ten[20], int ngay, int thang, int nam, char gioi[10], char noisinh[70],  
char nganh[30], int n, string tenmh[8], float diem[8], int tc[8], float dtb) {  
    MSSV = Id;  
    strcpy(hoten, ten);  
  
    dd = ngay;  
    mm = thang;  
    yyyy = nam;  
    strcpy(gioitinh, gioi);  
    strcpy(noisinh, noi);  
    strcpy(this->nganh, nganh);  
    for(int i=0; i<n; i++){  
        ten_mon_hoc[i] = tenmh[i];  
        mon_hoc[i] = diem[i];  
        so_tin_chi[i] = tc[i];  
    }  
    this->dtb = dtb;  
}
```

✓ Hàm tạo có tham số:

```

Sinhvien::Sinhvien(const Sinhvien& x) {
    MSSV = x.MSSV;
    dd = x.dd;
    mm = x.mm;
    yyyy = x.yyyy;
    strcpy(hoten, x.hoten);
    strcpy(gioitinh, x.gioitinh);
    strcpy(noisinh, x.noisinh);
    strcpy(nganh, x.nganh);
    size_mh=x.size_mh;
    for(int i=0;i<size_mh;i++){
        ten_mon_hoc[i]=x.ten_mon_hoc[i];
        mon_hoc[i]=x.mon_hoc[i];
        so_tin_chi[i]=x.so_tin_chi[i];
    }
    dtb=x.dtb;
}

```

✓ Hàm tạo sao chép:

## **B) HÀM HỦY**

```

Sinhvien::~Sinhvien() {}//////////hàm hủy

```



## C) TOÁN TỬ NHẬP XUẤT

### ➤ Toán tử nhập:

```
istream& operator>>(istream& is, Sinhvien& x) {
```

```
    cout<<"Ma so sinh vien la: ";
```

```
    is>>x.MSSV;
```

```
    is.ignore(1);
```

```
    cout<<"Ten la: ";
```

```
    is.getline(x.hoten,20);
```

```
    cout<<"Noi sinh la: ";
```

```
    is.getline(x.noisinh,70);
```

```
    cout<<"Gioi tinh la: ";
```

```
    is.getline(x.gioitinh,10);
```

```
    cout<<"Nganh la: ";
```

```
    is.getline(x.nganh,30);
```

```
    cout<<"Nhap ngay thang nam sinh la: " <<endl;
```

```
    cout<<"Ngay la: ";
```

```
    is>>x.dd;
```

```
    cout<<"Thang la: ";
```

```
    is>>x.mm;
```

```
    cout<<"Nam la: ";
```

```
    is>>x.yyyy;
```

```
    cout<<"So luong mon hoc can nhap diem la: ";
```

```
    is>>x.size_mh;
```

```
    for(int j=0;j<x.size_mh;j++){
```

```
        is.ignore(1);
```

```
        cout<<"Nhap ten mon hoc la: ";
```

```
        getline(is,x.ten_mon_hoc[j]);
```

```
        cout<<"Nhap diem la: ";
```

```
        is>>x.mon_hoc[j];
```

```
        cout<<"Tin chi la: ";
```

```
        is>>x.so_tin_chi[j];
```

```
    }
```

```
    x.dtb=diem_he_4(x.mon_hoc,x.so_tin_chi,x.size_mh);
```

```
    return is;
```

```
}
```

```

ostream& operator<<(ostream& os, Sinhvien& x) {
    os<< x.MSSV <<endl;
    os<< x.dd<<" " <<setw(2)<<x.mm<<" " <<setw(4)<<x.yyyy <<endl;
    os<< x.hoten <<endl;
    os<< x.gioitinh <<endl;
    os<< x.noisinh <<endl;
    os<< x.nganh <<endl;
    os<< x.size_mh <<endl;
    for(int i=0;i<x.size_mh;i++){
        os<< x.ten_mon_hoc[i] <<endl;
        os<< x.mon_hoc[i] <<endl;
        os<< x.so_tin_chi[i] <<endl;
    }
    os<< setprecision(4);
    os<< x.dtb <<endl;
    return os;
}

```

➤ Toán tử xuất:

```
int Sinhvien::getNgay(){
    return dd;
}
int Sinhvien::getThang(){
    return mm;
}
int Sinhvien::getNam(){
    return yyyy;
}
char Sinhvien::getTen(){
    return hoten;
}
char Sinhvien::getGT(){
    return gioitinh;
}
char Sinhvien::getNoiSinh(){
    return noisinh;
}
char Sinhvien::getNganh(){
    return nganh;
}
int Sinhvien::getMSSV(){
    return MSSV;
}
float Sinhvien::getdtb(){
    return dtb;
}
```

## D) CÁC HÀM KHÁC

- Hàm lấy giá trị:

- Toán tử so sánh >, <, ==

```
////////////////////toán tử so sánh >
```

```
bool Sinhvien::operator>(Sinhvien sv){
```

```
    if(DTB()>sv.DTB())
```

```
        return 1;
```

```
    return 0;
```

```
}
```

```
////////////////////toán tử so sánh <
```

```
bool Sinhvien::operator<(Sinhvien sv){
```

```
    if(DTB()<sv.DTB())
```

```
        return 1;
```

```
    return 0;
```

```
}
```

```
////////////////////toán tử so sánh ==
```

```
bool Sinhvien::operator==(Sinhvien sv){
```

```
    if(DTB()==sv.DTB())
```

```
        return 1;
```

```
    return 0;
```

```
}
```

```
////////////////////////toán tử so sánh >=
bool Sinhvien::operator>=(Sinhvien sv){
    if(DTB()>=sv.DTB())
        return 1;
    return 0;
}

////////////////////////toán tử so sánh <=
bool Sinhvien::operator<=(Sinhvien sv){
    if(DTB()<=sv.DTB())
        return 1;
    return 0;
}

////////////////////////toán tử so sánh !=
bool Sinhvien::operator!=(Sinhvien sv){
    if(DTB()!=sv.DTB())
        return 1;
    return 0;
}
```

- Toán tử so sánh >=,<=,!=

- Tính điểm trung bình hệ số 4

```
////////////////////////////////////////tính điểm trung bình hệ số 4
float diem_he_4(float diem[], int tin_chi[], int n){
    float b[n], Sum=0;
    int tch=0;
    for(int j=0;j<n;j++){
        if(diem[j]<=1.9) b[j]=0;
        else if(diem[j]>=2.0 && diem[j]<=3.9) b[j]=0.5;
        else if(diem[j]>=4.0 && diem[j]<=4.4) b[j]=1;
        else if(diem[j]>=4.5 && diem[j]<=5.4) b[j]=1.5;
        else if(diem[j]>=5.5 && diem[j]<=5.9) b[j]=2;
        else if(diem[j]>=6.0 && diem[j]<=6.9) b[j]=2.5;
        else if(diem[j]>=7.0 && diem[j]<=7.9) b[j]=3.0;
        else if(diem[j]>=8.0 && diem[j]<=8.4) b[j]=3.5;
        else if(diem[j]>=8.5 && diem[j]<=9.4) b[j]=3.8;
        else b[j]=4;
        tch+=tin_chi[j];
        Sum=Sum+b[j]*((float) tin_chi[j]);
    }
    float dtb=(float) Sum/tch;
    return dtb;
}
```



## 4) HÀM MAIN

```
int main(){
    Sinhvien a(),b();
    cout<< "Thong tin sinh vien la:" <<endl;
    cout<< a <<endl;
    cout<< b <<endl;
    cout<< "*****\n";
    cout<< "Toan tu so sanh >,<,<= " <<endl;
    if(a>b)
        cout<< "Diem tb a cao hon" <<endl;
    else if(a<b)
        cout<< "Diem tb a be hon b" <<endl;
    else if(a==b)
        cout<< "a=b" <<endl;
    else
        cout<< "khong xac dinh" <<endl;
    cout<< "*****\n";
    cout<< "Toan tu so sanh >= <=, != " <<endl;
    if(a>=b)
        cout<< "Diem tb a lon hon hoac bang b" <<endl;
    if(a<=b)
        cout<< "Diem tb a be hon hoac bang b" <<endl;
    if(a!=b)
        cout<< "a!=b" <<endl;
    system("pause");
    return 0;
}
```

Thank You,  
dreamstime  
Teacher

BÀI THUYẾT TRÌNH CỦA EM ĐẾN  
ĐÂY LÀ KẾT THÚC. XIN CẢM ƠN  
CÔ VÀ CÁC BẠN ĐÃ LẮNG NGHE.

