Ôn kỹ thuật lập trình hướng đối tượng

Những cái học thuộc là màu xanh dương.

Chú ý màu đỏ.

Xanh lá thì thay theo tùy bài

**A. Thư viện:**

Những thư viện sau hầu như là đều dùng.

Nhớ phân biệt được thì tốt, không thì cứ gõ hết vào.

#include<iostream.h>

// Khai báo để sử dụng được các lệnh nhập xuất thông thường: Cout, cin,.. cái này thì bài nào cũng phải dùng

#include<string.h>

//Khai báo để có thể nhập xâu kí tự, gọi các hàm xử lí xâu kí tự. Thường những bài nó cho nhập một cái gì kiểu char như tên, tuổi địa chỉ... thì nhớ khai báo cái này.

#include<iomanip.h>

// cái này thì để có những cái hàm xử lí cái in ra cho đẹp.. ví dụ dùng để gọi lệnh: setw(x) : in ra n dấu cách...... có cũng được ko thì mình dùng dấu cách đặt trong 2 dấu ngoặc kép “”

#include<math.h>

//Thư viện toán học khai báo để sử dụng các câu lệnh toán học như: abs(tuyệt đối), sqrt(căn bậc hai), sin, cos......

**B. Xây dựng lớp**

lớp nó có dạng sau này: cái xây dựng lớp thì 100% bài nào cũng phải có nó để thực hiện làm bài tập.

class <tên lớp>

{

private:

// Các thuộc tính.. nhớ các cái dữ liệu ở đây là thuộc tính.. chỗ này khai báo biên

protected:

// cái này gần tương tự private: nhưng không quen với không hiểu thì không nên dùng

tốt nhất dùng private thôi.

public:

// các thuộc tính như trên private:

// các phương thức... là những cái như hàm bình thường đấy như hàm nhập(), in()...... cái đấy gọi là phương thức... mấy hàm này thì trước học như nào thì làm như thế..

}; // Nhớ chỗ này chấm phẩy.

**I. Khai báo đúng và đủ các thuộc tính (0.5 điểm)**

Câu này đề bài nó cho gì thì mình dựa vào đấy để khai báo.

Thường khai báo ở private. Nếu có khai báo ở public thì nó cũng bảo.

cái này dễ cứ theo đề bài mà làm

ví dụ:

1. Xây dựng lớp Diem gồm các thuộc tính hoành độ x và tung độ y (x, y nguyên) là các thuộc tính riêng. Trong lớp có định nghĩa các hàm công cộng sau:

---------------

đây ví dụ như cái đầu bài như này.

Thì mình xác định như sau này:

Tên của class nó cho rồi là Diem==> class Diem{}

Thuộc tính là có 2 thuộc tính là x và y kiểu nguyên tức là: int == > int x, y;

mà nó ghi là thuộc tính riêng.. có nghĩa là private.

== > đủ cái để khai báo đúng và đủ thuộc tính rồi nhá.

cả đoạn trên sẽ như sau này:

class Diem

{

private:

int x;

int y;

// xong yêu cầu đầu tiên.. đoạn dưới này.. dưới học thêm sẽ viết thêm nhá.......

};

ví dụ 2:

Xây dựng lớp giáo viên GV gồm các thuộc tính họ tên ht (kiểu char[30]), chuyên ngành (kiểu char[20]), bậc lương bl (kiểu float) là các thuộc tính riêng. Trong lớp có định nghĩa các hàm công cộng sau:..........................

---------------

Bài nó cho như này thì cũng xác định tương tự như bài trên thôi

tên class: GV

Thuộc tính RIÊNG(private): ht, chuyên ngành, bl

#include<.........>

class GV

{

private:

char ht[30];

char chuyennganh[20];

float bl;

----------------------------------------------

};

ví dụ 3:

Xây dựng lớp phân số PS gồm 2 thuộc tính tử số ts và mẫu số ms (ts, ms nguyên) là các thuộc tính riêng.

class PS

{

private:

int ts;

int ms;

......................

};

ví dụ 4:

xây dựng 1 lớp sinh viên gồm họ tên (ht kiểu char[30], ngày sinh (ns), giới tính(gt kiểu char[4]), lớp(char [20]), điểm toán t, lí l, hóa h, trung bình tb (t, l, h, tb kiểu float)

Riêng đối với bài này... nhỡ may gặp thôi

nó có cái ngày sinh ý... thì ngày sinh nó dạng: ngày/tháng/ năm

mà cái kiểu ngày thì c++ nó không có sẵn mà do mình tự định nghĩa

nên cái này sẽ học thuộc nhỡ nó vào.

struct NS

{

int ngay, thang, nam;

};

đấy cái trên là tạo ra 1 cái kiểu ngày tháng năm...

thì sau nó sẽ thành 1 kiểu như kiểu có sẵn int, float, char.

Chú ý là nó phải khai báo và tạo trước class nhá.

Xong thì khi gọi biến cho ngày sinh, thì gọi như sau này:

NS nsinh; // NS là tên cái struct mình đặt, đặt là gì thì giờ gọi lại là thế.

còn cái nsinh thì là tên biến thôi.. muốn đặt là gì thì đặt. Nhìn cái này nó cũng giống như khai báo biến bình thương thôi... ví dụ: int x; float y.. thì đằng trước là tên kiểu dữ liệu.. đằng sau

là tên biến.

còn khi mà mình truy nhập thì dùng cái tên đằng sau của lúc khai báo: NS nsinh... thì muốn nhập cho ngày thì viết : nsinh.ngay; cho tháng thì ns.thang... đấy là gọi đến thành phần của struct. Nhớ cái này nhá.

#include....

struct NS

{

int ngay, thang, nam;

};

class SV

{

private:

char ht[30];

NS nsinh;

char gt[4];

char lop[20];

float t, l, h, tb;

............................................

};

XONG Ý ĐẦU TIÊN... KHAI BÁO ĐÚNG VÀ ĐỦ CÁC THUỘC TÍNH (0,5 điểm nhá)... Cái này dễ nhá.

II.

a) Khai báo đúng và đủ các phương thức.(0,5 điểm)

b) Định nghĩa các phương thức(tầm 2.5 điểm)

Hàm khởi tạo không và có tham số(1 điểm)//chắc chắn có vs mọi bài

Hàm hủy (0,5)// chắc chắn có với mọi bài.

Nhập (0,5điểm)// hầu như đều có

Xuất(0,5 điểm)// hầu như đều có

// có thể có 1 số yêu cầu khác.......

Phần này nói túm hết 1 phát luôn nhá.. Cái phần này nhìn nó nhiều thôi nhưng toàn cái dễ hiểu, dễ làm, dễ biết, cũng như dễ thuộc dễ thay... Mà tổng hết vào là tầm 5 điểm rồi..

cố mà học cái này cho khỏi chết....

Cái này thì tương tự như cái thuộc tính thôi.

Nhưng nhớ là phương thức là những cái nó giống kiểu như hàm ý

và hầu như đều khai báo ở public nhá..

Có phương thức nào đầu bài nó thường bắt luôn.

Nên cứ thế mà làm.

Nhớ là có hai cái thì bài nào cũng phải có nhá đó là HÀM TẠO VÀ HÀM HỦY.

hai hàm này thì cực dễ luôn...:D

khai báo phương thức theo cấu trúc này này:

<kiểu dữ liệu trả về> <tên phương thức>

{

// các lệnh bên trong...

}

ví dụ:

void nhap()

{

cout<<” Xin chao cac ban”;

cin>>x;

}

Nó giống như một hàm thông thường thôi.

Giờ thì nói lần lượt nhá.

Đầu tiền là hàm tạo.

hàm tạo trong bài thì mình phải tạo ra 2 cái hàm tạo. bài nào cũng thế. đó là hàm tạo không đối và có đối.

cấu trúc như sau này:

hàm tạo không đối:

<tên class>()

{

//khởi tạo các biến

}

hàm tạo có đối

<tên class> (các đối trong này này.. có bao nhiêu thuộc tính ở cái private mới học bên trên ý, thì ở đây truyền vào từng ý)

{

// khởi tạo các biến

}

ví dụ vẫn mấy bài ở trên nhá cho liền mạch:

1. Xây dựng lớp Diem gồm các thuộc tính hoành độ x và tung độ y (x, y nguyên) là các thuộc tính riêng. Trong lớp có định nghĩa các hàm công cộng sau:

* Hàm khởi tạo với các tham số có giá trị ngầm định bằng 0

#include<.....

class Diem

{

private:

int x;

int y;

public:

// Hàm tạo bắt đầu này.. nhìn so với cấu trúc ở trên nhá.

//Tạo không đối

Diem()

{

// trong này có thuộc tính(biến trong private) nào thì cứ thế mà mang xuống gán giá trị thôi.........

x=0;

y=0;

}

// Tạo thêm cả 1 cái có đối nữa.

// Cái này thì tương tự. ở trên có gì thì ở dưới lôi hết vào trong dấu ngoặc tròn ý.. xong thêm số 1 vào cuối tên biến(thêm j cug được ko nhất thiết là 1). Xong bên trong {} thì mình lấy tến biến trên private: gán lần lượt cho các cái trong ngoặc tròn ở hàm tạo() ... ok

Diem(int x1, int y1)

{

x=x1;

y=y1;

}

...........................................

};

thế là xong hàm tạo.. Giờ là một số cái dạng khác nữa cho đủ nhá.

ví dụ 2: