1. THƯ VIỆN

void main()

{

char x = 'H';

string z = "Anh thang dep trai";

cout << x << " ;";

cout<<z;

muoinamsau("Anh thang xau trai");

\_getch();

}

NHỚ KHAI BÁO THƯ VIỆN : #include<string>

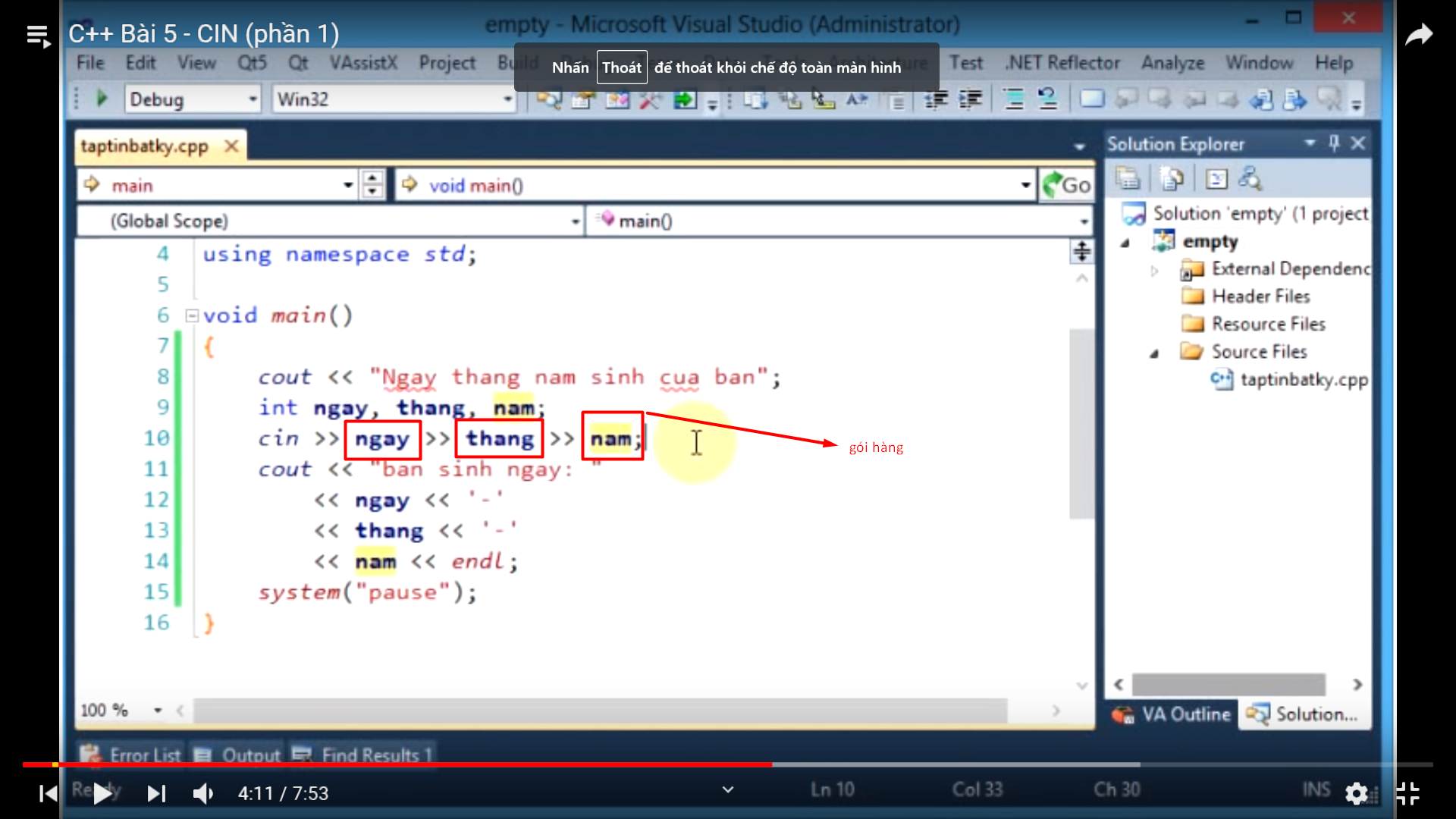
1. THƯ VIỆN

#include<conio.h> 🡪 \_getch(); OR SYSTEM(“PAUSE”)

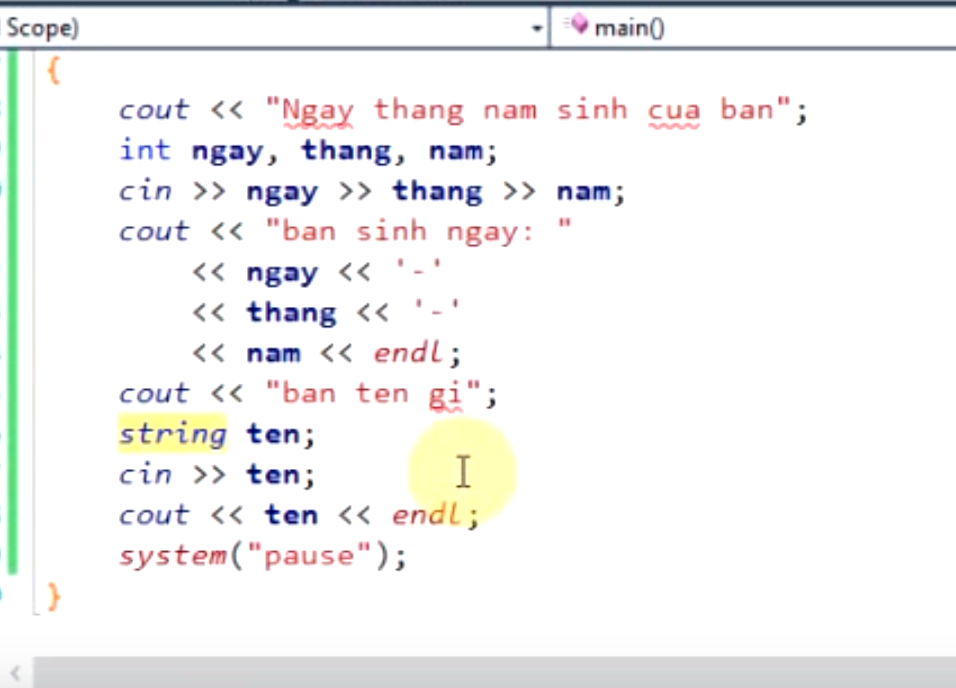
1. CIN ----- MỖI CÁI CIN GIỮ MỘT “GÓI HÀNG” TRUYỀN VÀO

Cin nhận từng gói

Và phân biệt từng gói là 1 dấu khoảng trắng



NẾU “GÓI HÀNG” TA NHẬP VÀO BỊ DƯ : VÍ DỤ 3 GÓI ->NHẬP 4 THÌ GÓI CUỐI CÙNG SẼ DC CHUYỂN VÀO CIN TIẾP THEO

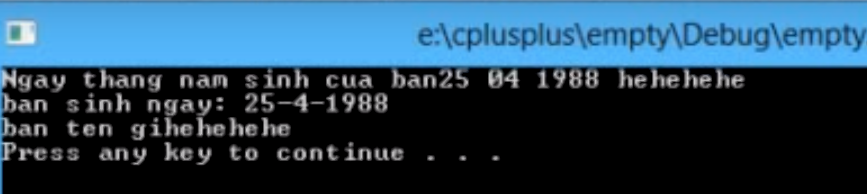


LẦN nhập 1: chỉ chứa 3 gói

Lần nhập 2: chứa 1 gói

CHẠY CONSOLE :NHẬP lần 1 : 25 - 04 – 1988 – hehehehe : 4 gói thì gói cuối cùng tự động nhảy xuống “cin>>tSen” lun!!!

Nghĩa là thay vì 2 lần nhập thì lần nhập thứ 2 đã tự động nhập được nhập là :”hehehehe”…

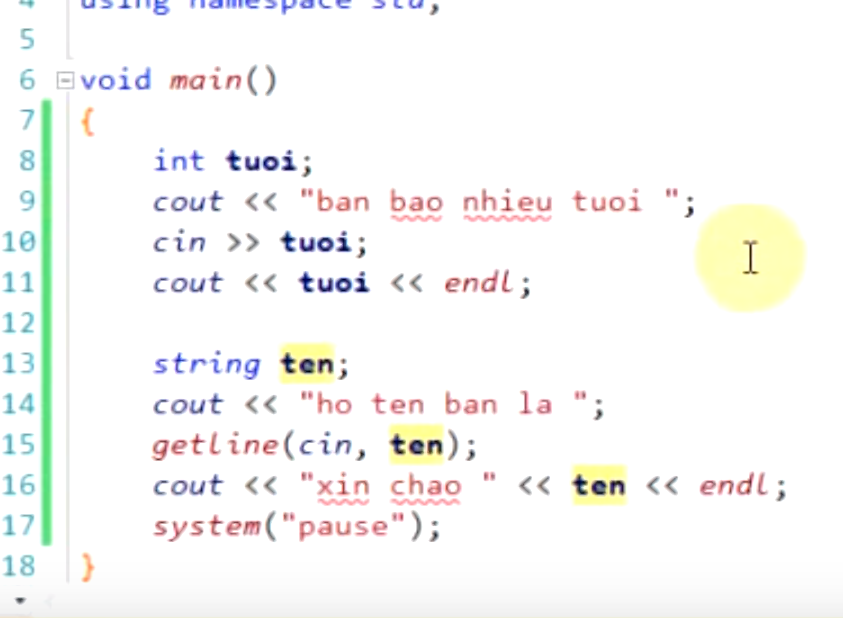


1. CIN (tiếp theo) :nhận lun dấu ENTER

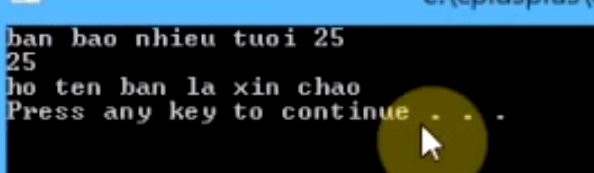
LƯU Ý :KHI DÙNG STRING THÌ PHẢI DÙNG GETLINE(X,Y)

X: LÀ GIÁ TRỊ CHUỖI ĐƯA VÀO CIN

Y: LÀ GIÁ TRỊ LƯU LẠI TRONG STRING



CHẠY CONSOLE: NHẬP VÀO 25 BẤM ENTER : MÁY NÓ NGU NÊN NÓ TƯỞNG ENTER LÀ GIÁ TRỊ NÊN NÓ HIỂU LÀ NHẬP 2 “GÓI HÀNG”

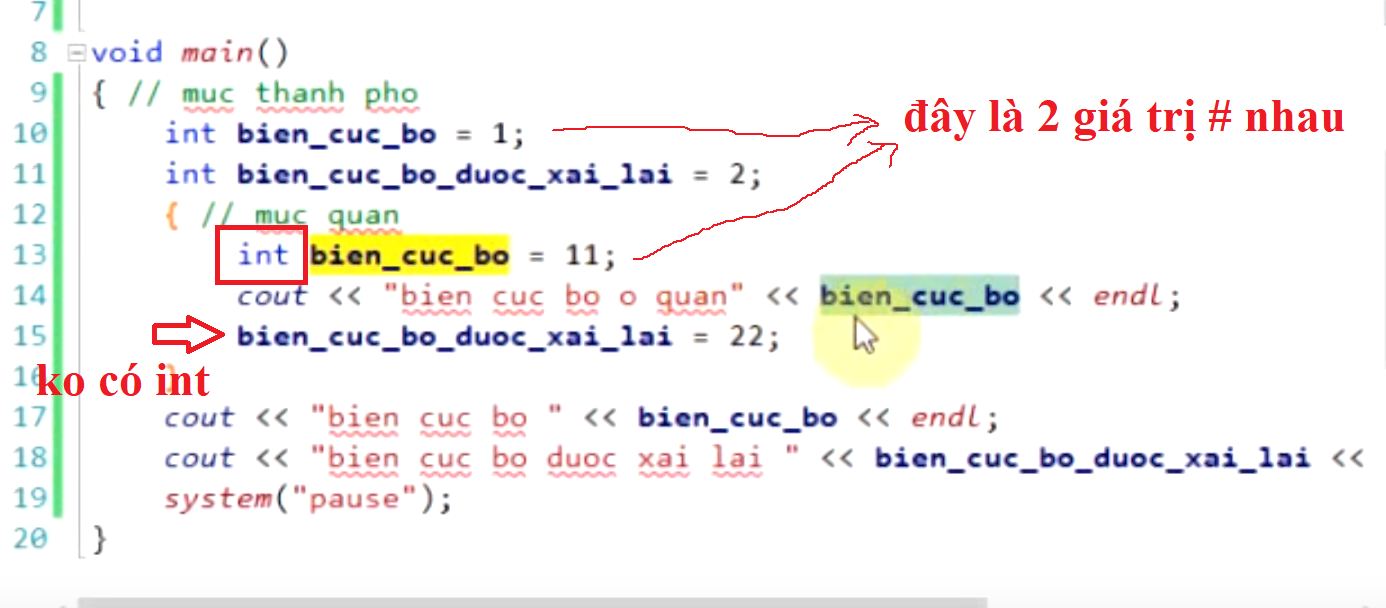


* KHÁC PHỤC DÙNG :

+ fflush(stdin); 🡪xóa enter

+ cin.ignore(x),🡪 x số kí tự bỏ qua

1. Nhớ lại tầm vực
2. BIẾN CỤC BỘ VÀ TOÀN CỤC



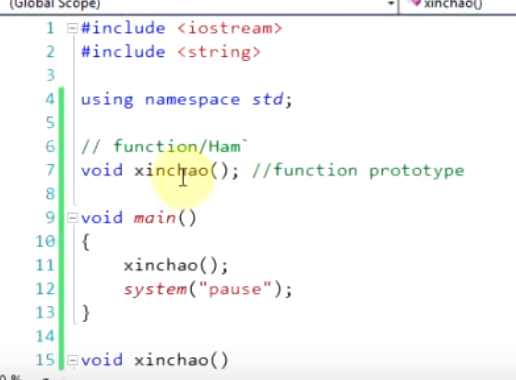
int bien\_cuc\_bo =1;

int bien\_cuc\_bo\_duoc\_xai\_lai=2;

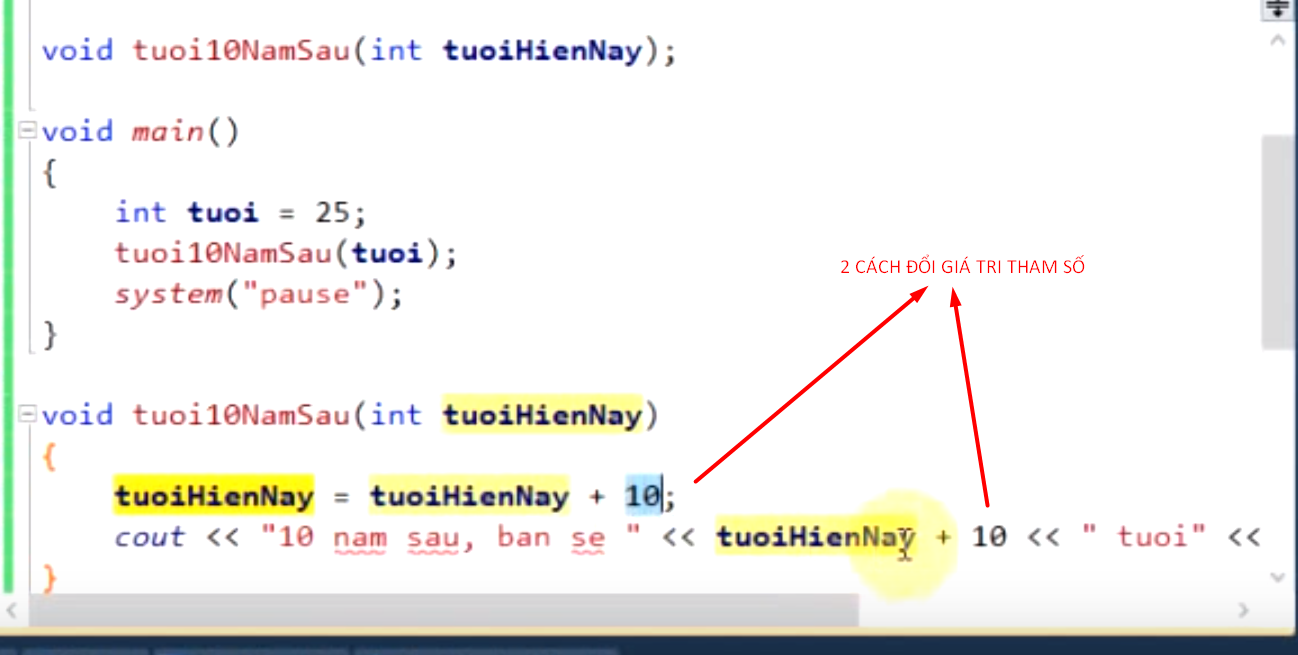
2 giá trị trên thì giá trị nào sẽ bị thay đổi ?

Da :là bien\_cuc\_bo\_duoc\_xai\_lai bị thay đổi

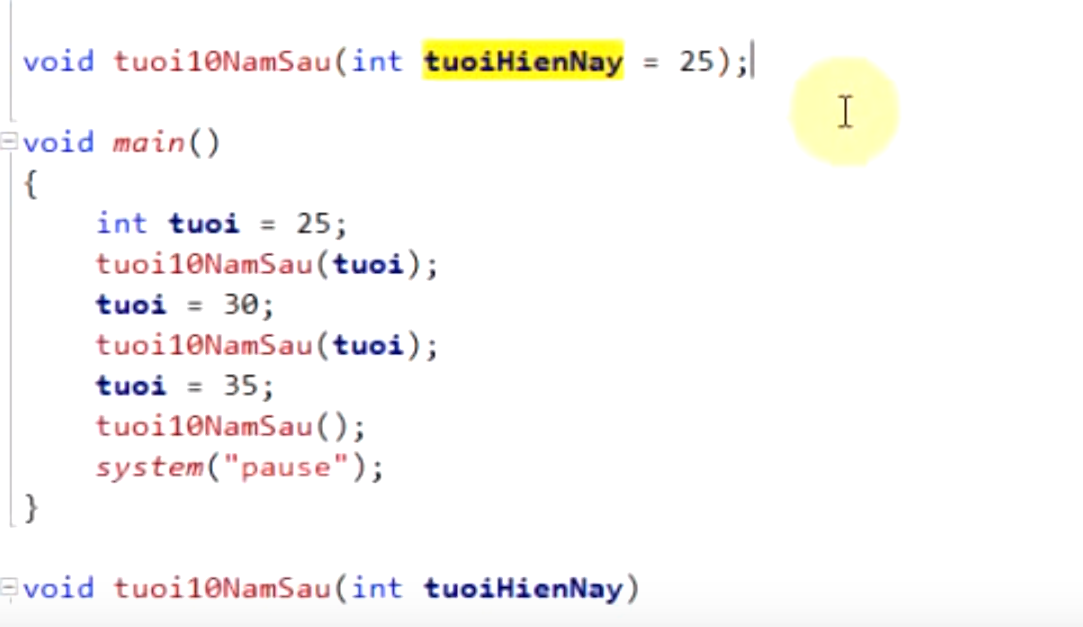
1. Hàm void cơ bản, biến, gán, if - else



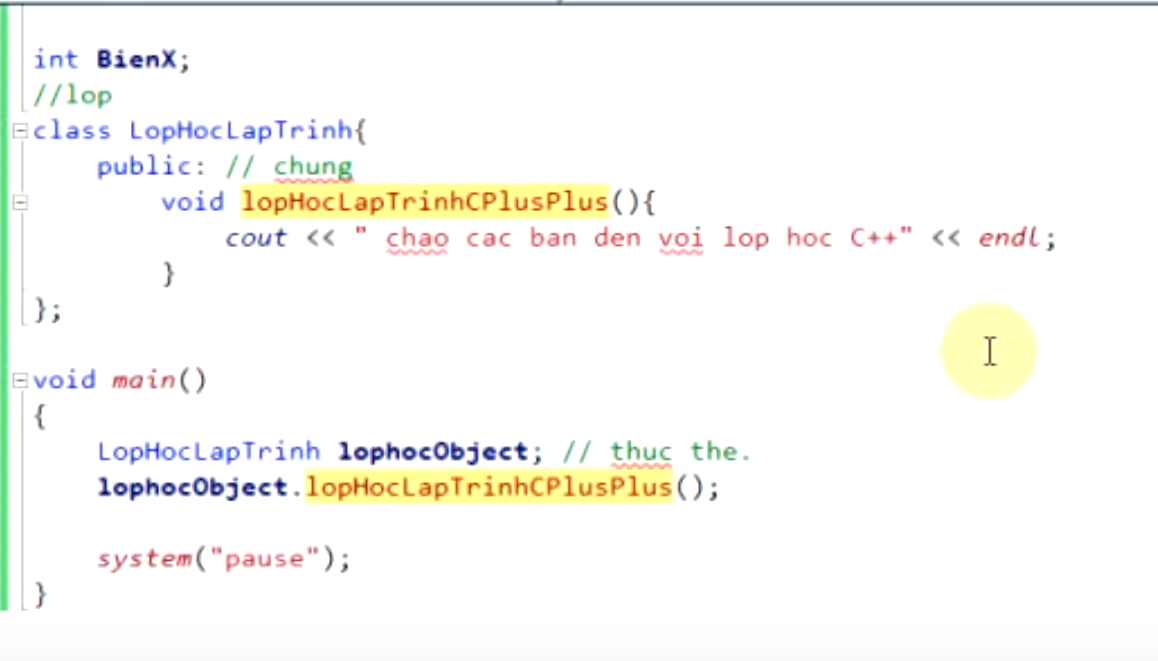
1. HÀM VOID TRUYỀN THAM SỐ

1: THAY ĐỔI THAM SỐ TRONG BODY (THÂN HÀM) 

2: THAY ĐỔI THAM SỐ NGAY MIỆNG HÀM (LƯU Ý CHỈ THAY ĐỔI Ở HÀM KHAI BÁO ) BẰNG CÁCH THỂ GÁN

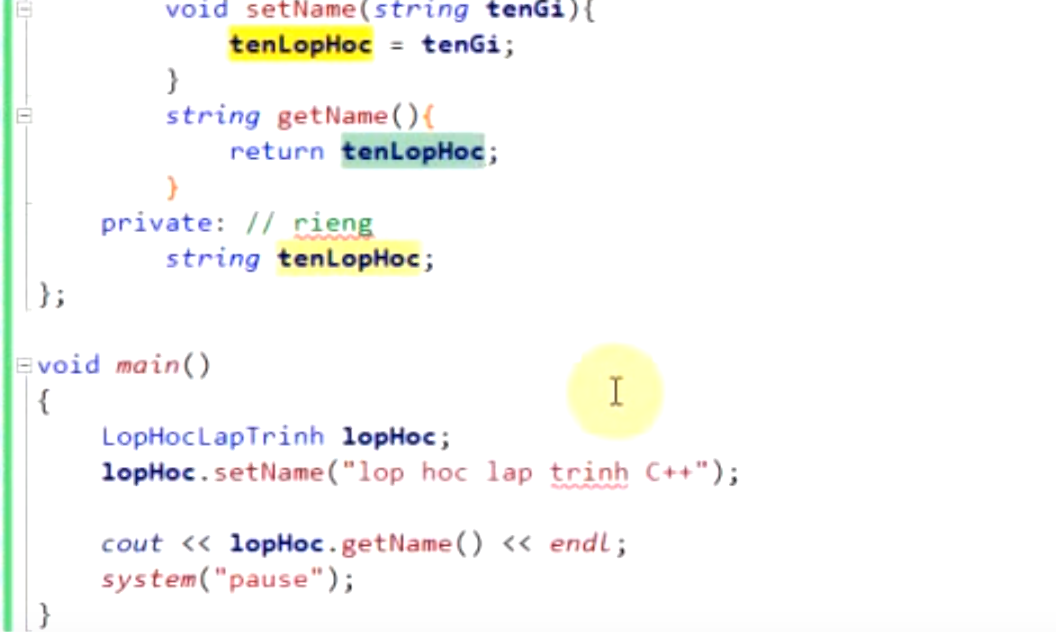


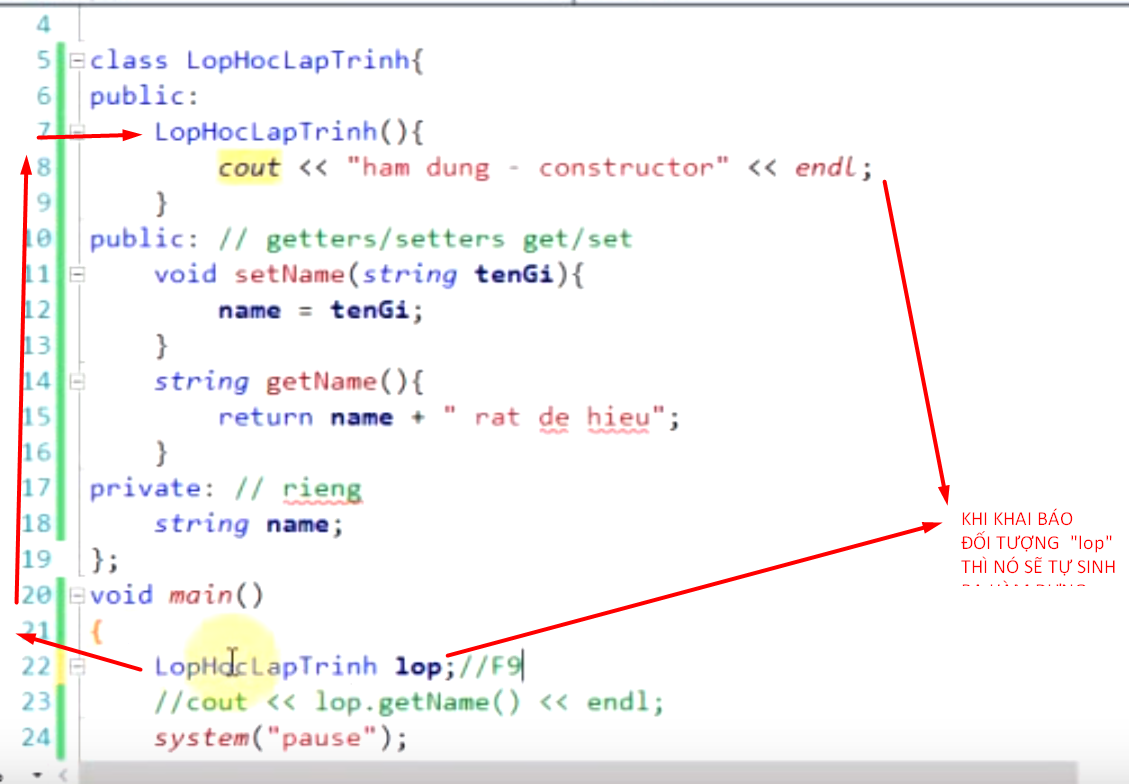
1. (QUAN TRỌNG) :LỚP



1. :LỚP (TIẾP THEO) VÀ CÁCH SỬ DỤNG BIẾN TRONG LỚP BẰNG GET/SET

GET/SET RẤT CÓ LỢI VỀ SAU – PHẢI HIỂU ĐIỀU NÀY





1. Con trỏ (cấp 1)

Giả sử :cho 2 biến a và b với a =3; b=0

-Yêu cầu gán b cho a để a =0 (bằng con trỏ)

void main()

{

int \*prA;

int a = 3;

int b = 0;

prA = &b;

a = \*prA;

cout << \*prA; // xuất ra giá trị con trỏ đang mang

cout << prA;// xuất ra địa chỉ con trỏ

cout << &a; //xuất ra địa chỉ biến

}

## LƯU Ý 1 :CÓ 2 CÁCH GÁN CON TRỎ (rất dễ nhầm)

int \*prA=&a; hoặc int \*prA; prA=&a

## LƯU Ý 2: Phân biệt 2 cái rất dễ nhầm trong con trỏ : địa chỉ và giá trị (xem chú thích con trỏ cấp 3)🡪 cả 2 cái này đều xuất ra 1 kết quả (nhưng khác nhau)

1. Con trỏ-đa cấp (nâng cao)= POINT TO POINT (QUAN TRỌNG)

RẤT DỄ : PHÂN BIỆT :

INT \*PT; 🡪 CON TRỎ CẤP 1

INT \*\*PT; 🡪 CON TRỎ TRỎ ĐẾN CON TRỎ (POINT –TO– POINT) (PTP)

LƯU Ý:

INT \*PT;

INT \*\*PT2;

GÁN ĐỊA CHỈ CHO CON TRỎ CẤP 1: PT =&A

GÁN ĐỊA CHỈ CHO CON TRỎ ĐA CẤP: PT2=&A🡪 ~~PT2=&A~~ Sai (vì con trỏ đa cấp ko đc gán một địa chỉ thường mà phải là địa chỉ của một con trỏ khác vì nó ko cùng giá trị kiểu (int \*\*)  ví dụ : PT2=&PT;

GÁN GIÁ TRỊ (POINT –TO– POINT) (PTP) : \*\*PT2 = \*(&A)

# **TÓM LẠI (QUAN TRỌNG): CON TRỎ ĐA CẤP(p\_t\_p) GÁN “ĐỊA CHỈ THƯỜNG” KO ĐC NHƯNG GÁN “GIÁ TRỊ THƯỜNG“ THÌ ĐC**

Còn \*pt2: là địa chỉ của mà con trỏ con lưu

VÍ DỤ : CON TRỎ ĐA CẤP (POINT –TO– POINT):

void main()

{

int a = 3;

int b = 0;

int \*pta;

int \*\*ptb;

int \*\*\*ptc;

pta = &a;

ptb = &pta;

ptc = &ptb;

cout << &a << endl;

cout << ptb << endl;

cout << pta<<endl;

cout << \*\*ptc << endl;

\*\*ptc = &b;//địa chỉ gán cho địa chỉ

cout << \*\*\*ptc << endl;;

\*\*\*ptc = \*(&b);//giá trị gán cho giá trị

cout << \*\*\*ptc << endl;;

\_getch();

}

1. CÁC CẤP ĐỘ CỦA 1 CON TRỎ ĐA CẤP

**int** **main**() {

**int** value = 100;

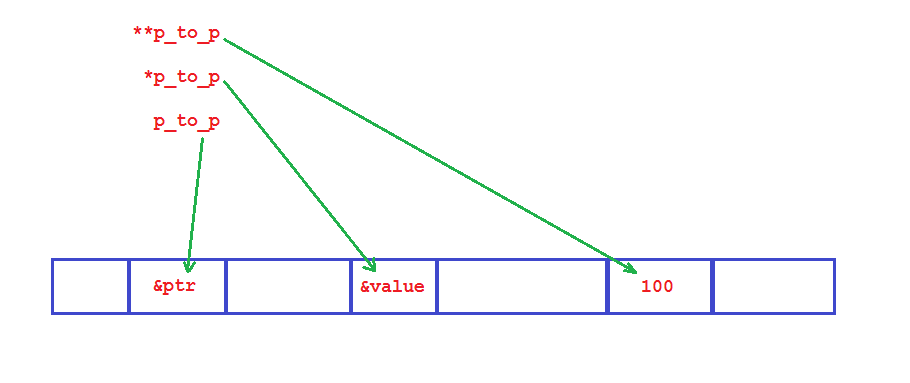
**int** \*ptr = &value;

**int** \*\*p\_to\_p = &ptr;

cout << p\_to\_p << endl; *//print address of ptr*

cout << \*p\_to\_p << endl; *//print address which hold by ptr*

cout << \*\*p\_to\_p << endl; *//print value at address which hold by ptr*

**return** 0;

\*\*p\_to\_p: xuất ra giá trị mà con trỏ đang giữ

\*p\_to\_p: lấy ra địa chỉ mà con trỏ cấp dưới nó đang giữ (tức là địa chỉ của mà value nó lưu gtri 100) ko phải địa chỉ của con trỏ cấp dưới con nha

p\_to\_p:cái này mới lấy địa chỉ con trỏ cấp dưới

i**nt** \*\*\*pt3; Pointer to pointer to pointer : Tuy nhiên, việc thao tác với những con trỏ như thế này khá phức tạp và rất ít gặp trong thực tế

1. LƯU Ý: BẢN CHẤT CỦA MẢNG (ARRAY) LÀ MỘT CON TRỎ.
2. Hàm random

#include <time.h>

void main()

{

*srand*(*time*(0)); // hàm rand() là hàm mặc định của chương trình nên ko chuyền tham số vào đc nếu muốn chuyền tham số phải dùng hàm srand(x);

*cout* << *rand*()%100; // lấy hàng chục

*\_getch*();

}