

# KỸ THUẬT LẬP TRÌNH C/C++

## Chương 8: KIỂU CẤU TRÚC

# Nội dung

1. Khái niệm kiểu cấu trúc
2. Khởi tạo cấu trúc
3. Cấu trúc lồng nhau

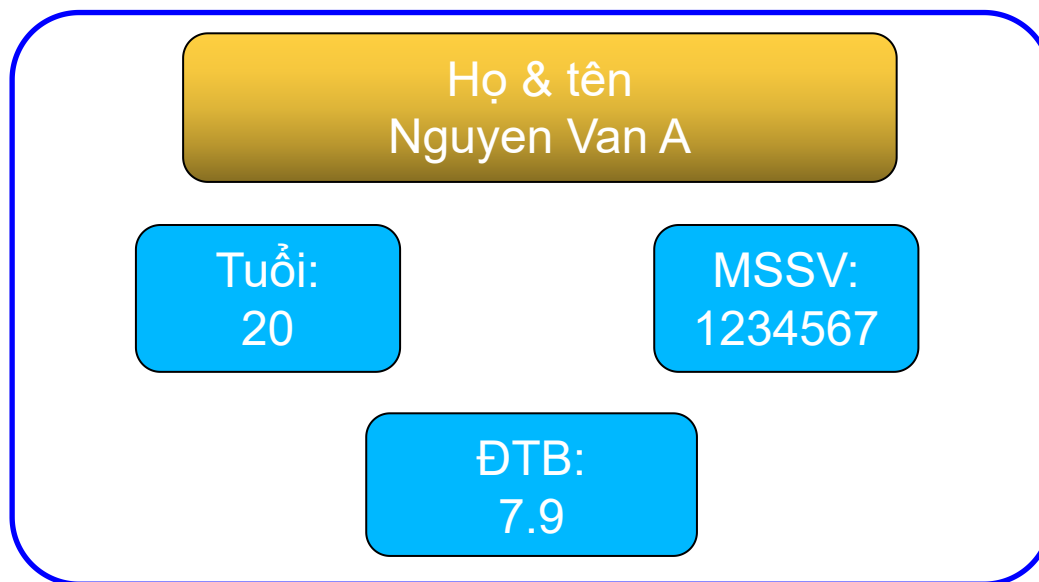
# 1. Khái niệm kiểu cấu trúc

# 1. Khái niệm kiểu cấu trúc

- Kiểu cấu trúc trong C/C++ là một tập hợp các thuộc tính liên quan tới cùng một đối tượng.

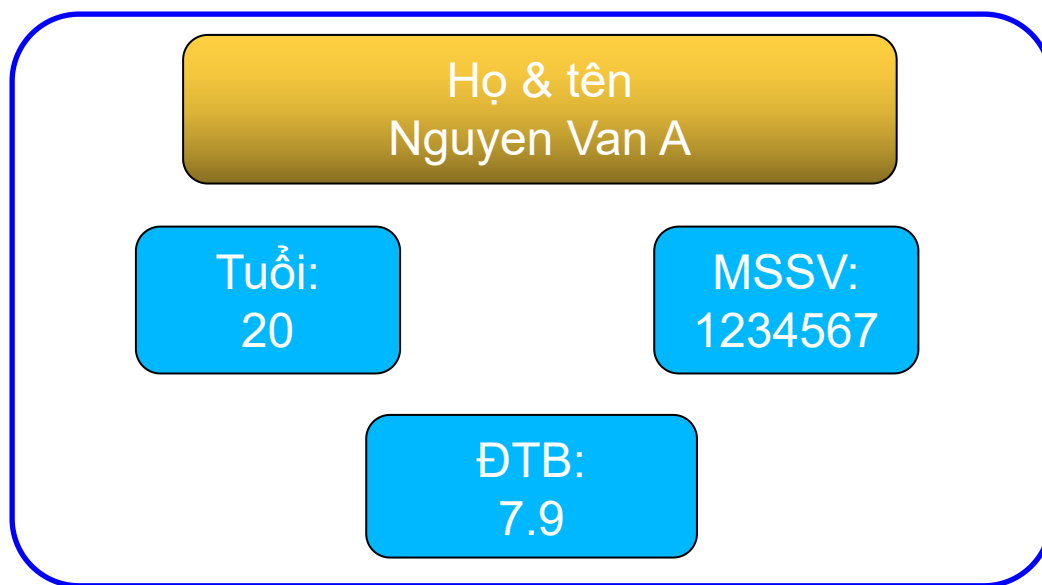
Ví dụ tập hợp các thuộc tính liên quan tới một sinh viên gồm: họ tên, tuổi, MSSV, điểm trung bình (ĐTB) => gọi là kiểu cấu trúc [sinh viên].

- Các thuộc tính tạo nên một cấu trúc được gọi là các thành viên (member) của cấu trúc đó. Trong ví dụ trên, “họ tên”, “tuổi”, “MSSV”, “ĐTB” chính là các thành viên trong cấu trúc. Tuy nhiên, chúng ta không thể sử dụng hàm làm thành viên của một cấu trúc được.



# 1. Khái niệm kiểu cấu trúc

- Từ một cấu trúc, chúng ta có thể tạo ra nhiều các thực thể (instance) có cấu trúc giống nhau. Từ một thực thể cụ thể, chúng ta có thể gọi các thành viên trong thực thể đó, cũng như gán các giá trị và sử dụng chúng.
- Với cấu trúc [sinh viên] trong ví dụ trên, chúng ta có thể tạo ra một thực thể là [Nguyen Van A], với các thành viên có giá trị cụ thể như sau:



# 1. Khái niệm kiểu cấu trúc

- Kiểu cấu trúc có khả năng nhóm các dữ liệu với nhiều kiểu dữ liệu khác nhau.

Với cấu trúc [sinh viên] trong ví dụ trên, chúng ta thấy rằng có 2 kiểu dữ liệu khác nhau như: kiểu chuỗi (họ tên), kiểu số nguyên (Tuổi, MSSV) và kiểu số thực (ĐTB).

- Ưu điểm lớn nhất của kiểu cấu trúc trong C/C++ đó chính là khả năng lưu trữ các dữ liệu thuộc nhiều kiểu dữ liệu khác nhau. Bằng việc sử dụng kiểu cấu trúc, các biến cùng liên quan tới một đối tượng trong chương trình với nhiều kiểu dữ liệu khác nhau có thể được kết hợp thành một khối, giúp chúng ta dễ dàng quản lý chúng trong chương trình cũng như giúp việc viết code đơn giản và ngắn gọn hơn.

## 2. Khởi tạo cấu trúc

- ❖ **Sử dụng từ khoá struct:** Một cấu trúc do người lập trình tự định nghĩa được khai báo thông qua từ khoá **struct** theo cú pháp như sau:

```
struct <Tên cấu trúc>{  
    <Kiểu dữ liệu 1> <Tên thuộc tính 1>;  
    <Kiểu dữ liệu 2> <Tên thuộc tính 2>;  
    ...  
    <Kiểu dữ liệu n> <Tên thuộc tính n>; };
```

Trong đó:

**struct:** là tên từ khoá để khai báo một cấu trúc, bắt buộc phải có khi định nghĩa cấu trúc.

**Tên cấu trúc:** là tên do người lập trình tự định nghĩa, tuân thủ theo quy tắc đặt tên biến trong C++.

**Tên thuộc tính:** mỗi thuộc tính của cấu trúc được khai báo như khai báo một biến thuộc kiểu dữ liệu thông thường và phải kết thúc bằng dấu chấm phẩy “;”

## 2. Khởi tạo cấu trúc

### ❖ Sử dụng từ khoá struct:

Ví dụ: Về việc quản lý sinh viên, khi xử lý thông tin về mỗi sinh viên, chúng ta luôn phải xử lý các thông tin liên quan như: “họ tên”, “tuổi”, “MSSV”, “ĐTB”. Do đó, chúng ta sẽ dùng cấu trúc để lưu giữ thông tin về mỗi sinh viên bằng cách định nghĩa một cấu trúc có tên là [sinhvien] với các thuộc tính như sau:

```
struct sinhvien{  
    string ten; //Họ tên sinh viên  
    int tuoi;   //Tuổi sinh viên  
    int mssv;   //Mã số sinh viên  
    float dtb; //Điểm trung bình  
};
```

Trong đó, sinhvien là một cấu trúc chứa 4 thành viên (4 biến) gồm: ten, tuoi, mssv, dtb.

Khi cấu trúc được khai báo, không có bộ nhớ được cấp phát cho các biến. Chỉ khi biến của cấu trúc được tạo, thì bộ nhớ mới được cấp phát cho các biến.



## 2. Khởi tạo cấu trúc

### ❖ Sử dụng từ khoá struct:

Ví dụ: Về việc quản lý sinh viên (tt)

```
#include <iostream>
#include <string.h>
using namespace std;

struct sinhvien{
    string ten; //Họ tên sinh viên
    int tuoi;   //Tuổi sinh viên
    int mssv;   //Mã số sinh viên
    float dtb;  //Điểm trung bình
};
```

//Hoặc có thể lưu trữ thông tin sv2 theo cách như sau:

```
struct sinhvien sv2 {"Nguyen Thi B", 21, 123123, 8.5};
```

```
int main() {
    //khai báo hai biến sv1, sv2 có kiểu cấu trúc sinhvien
    struct sinhvien sv1, sv2;

    //lưu trữ thông tin của sv1
    sv1.ten = "Nguyen Van A";
    sv1.tuoi = 20;
    sv1.mssv = 1234567;
    sv1.dtb = 7.9;

    //lưu trữ thông tin của sv2
    sv2.ten = "Nguyen Van B";
    sv2.tuoi = 21;
    sv2.mssv = 123123;
    sv2.dtb = 8.5;

    //truy xuất thông tin sinh viên
    cout << "Ho ten sv1 la:" <<sv1.ten <<endl;
    cout << "DTB cua sv1 la:" <<sv1.dtb <<endl;
    cout << "Ho ten sv2 la:" <<sv2.ten <<endl;
    cout << "DTB cua sv2 la:" <<sv2.dtb <<endl;
}
```

## 2. Khởi tạo cấu trúc

### ❖ Sử dụng từ khoá typedef:

Để tránh phải dùng từ khoá struct mỗi khi khai báo biến cấu trúc, ta có thể dùng từ khoá typedef khi định nghĩa cấu trúc theo cú pháp như sau:

```
typedef struct {  
    <Kiểu dữ liệu 1> <Tên thuộc tính 1>;  
    <Kiểu dữ liệu 2> <Tên thuộc tính 2>;  
    ...  
    <Kiểu dữ liệu n> <Tên thuộc tính n>;  
} <Tên kiểu dữ liệu cấu trúc>;
```

Trong đó:

<Tên kiểu dữ liệu cấu trúc>: là tên kiểu dữ liệu của cấu trúc vừa định nghĩa. Tên này sẽ được dùng như một kiểu dữ liệu thông thường khi khai báo biến cấu trúc.

## 2. Khởi tạo cấu trúc

### ❖ Sử dụng từ khoá typedef:

Tương tự ví dụ về việc quản lý sinh viên, với các thông tin liên quan như: “họ tên”, “tuổi”, “MSSV”, “ĐTB”. Khởi tạo cấu trúc bằng cách sử dụng từ khoá **typedef**. Tương tự code như trên nhưng chỉ thay đổi như sau:

...

**typedef struct**{

    string ten; //Họ tên sinh viên

    int tuoi; //Tuổi sinh viên

    int mssv; //Mã số sinh viên

    float dtb; //Điểm trung bình

**}sinhvien;**

int main() {

    //khai báo hai biến sv1, sv2 có kiểu cấu trúc sinhvien

**sinhvien sv1, sv2;**

    //lưu trữ thông tin của sv1

    ...

    //lưu trữ thông tin của sv2

    ...

    //truy xuất thông tin sinh viên

}

### 3. Cấu trúc lồng nhau

Trong ví dụ về quản lý sinh viên ở trên, bây giờ chúng ta không quan tâm đến tuổi sinh viên nữa, mà quan tâm đến ngày sinh của sinh viên. Vì ngày sinh cần có các thông tin luôn đi với nhau là ngày sinh, tháng sinh, năm sinh. Do đó, ta định nghĩa một kiểu cấu trúc con cho kiểu ngày sinh:

```
struct Date{
    int day; //Ngày sinh
    int month; //Tháng sinh
    int year; //Năm sinh
};

struct sinhvien{
    string ten; //Họ tên sinh viên
    Date ntns; //Ngày tháng năm sinh
    int mssv; //Mã số sinh viên
    float dtb; //Điểm trung bình
};

int main() {
    //khai báo hai biến sv1, sv2 có kiểu cấu trúc sinhvien
    struct sinhvien sv1, sv2;
    //lưu trữ thông tin của sv1
    sv1.ten = "Nguyen Van A";
    sv1.ntns.day = 15;
    sv1.ntns.month = 9;
    sv1.ntns.year = 2003;
    sv1.mssv = 1234567;
    sv1.dtb = 7.9;
    //truy xuất thông tin sinh viên
    cout << "Ho ten sv1 la:" <<sv1.ten <<endl;
    cout << "Ngay sinh cua sv1 la:" <<sv1.ntns.day <<endl;
    cout << "Thang sinh cua sv1 la:" <<sv1.ntns.month <<endl;
    cout << "Nam sinh cua sv1 la:" <<sv1.ntns.year <<endl;
}
```