

# Ngôn ngữ lập trình C

## B14: Cấu trúc (Structure)

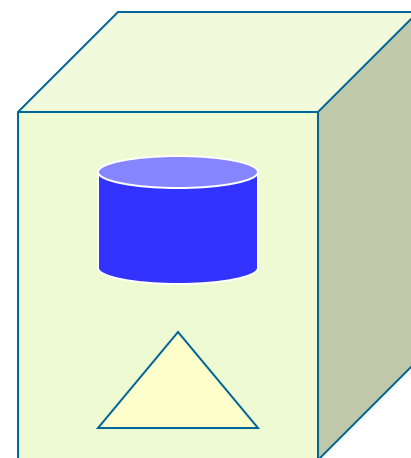


**PHENIKAA**  
UNIVERSITY

**Khoa Công nghệ thông tin**

# Khái niệm cấu trúc

- Kiểu dữ liệu cấu trúc (`struct`) là kiểu dữ liệu phức hợp bao gồm nhiều thành phần, mỗi thành phần có thể thuộc những kiểu dữ liệu khác nhau.
- Các thành phần dữ liệu trong cấu trúc được gọi là các trường dữ liệu (`field`).



# Khái niệm cấu trúc

- Ví dụ: Cấu trúc mô tả sinh viên:
  - Cần lưu giữ các thông tin liên quan đến sinh viên như:
    - Họ tên,
    - Tuổi,
    - Kết quả học tập, ...
  - Mỗi thông tin thành phần lại có kiểu dữ liệu khác nhau như:
    - Họ tên có kiểu dữ liệu là xâu kí tự,
    - Tuổi có kiểu dữ liệu là số nguyên,
    - Kết quả học tập có kiểu dữ liệu là số thực.

# Khai báo kiểu dữ liệu cấu trúc

- Cú pháp:

```
struct ten_cau_truc
{
    <khai_bao_cac_truong_du_lieu>;
};
```

- Ví dụ:

```
struct sinh_vien
{
    char ma_so_sinh_vien[10];
    char ho_va_ten[30];
    float diem_thi;
};
```

# Khai báo biến cấu trúc

- Cú pháp:

```
struct ten_cau_truc ten_bien_cau_truc;
```

- Ví dụ:

```
struct sinh_vien a, b, c;
```

```
struct car {
```

```
    char* make;
```

```
    char* model;
```

```
    int    year;
```

```
};
```

```
struct car mycar;
```

```
struct sinh_vien s1 = {"SV123", "Nguyen Van A", 8.0};
```

```
struct car oto1 = {"Toyota", "Vios", 2019};
```

# Khai báo kết hợp

- Có thể kết hợp vừa khai báo kiểu dữ liệu cấu trúc vừa khai báo biến cấu trúc:

```
struct [ten_cau_truc]
{
    //khai_bao_cac_truong;
} ten_bien_cau_truc;
```

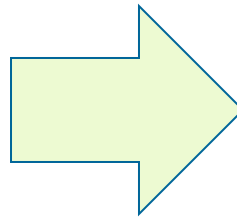
- Ví dụ:

```
struct thi_sinh
{
    char SBD[10]; // so bao danh
    char ho_va_ten[30];
    struct diem_thi ket_qua;
} thi_sinh_1, thi_sinh_2;
```

# Định nghĩa kiểu dữ liệu cấu trúc với `typedef`

- Đặt tên mới cho kiểu dữ liệu cấu trúc.
- Cú pháp: `typedef struct tên_cũ tên_mới;`
- Lưu ý: được phép đặt tên\_mới trùng với tên\_cũ.

```
typedef struct student {  
    char name[20];  
    int age;  
    float grade;  
} student_t;
```



Chương trình sẽ có  
kiểu dữ liệu mới là:  
**student\_t and car\_t**

```
typedef struct car {  
    char* make;  
    char* model;  
    int year;  
} car_t;
```

# Định nghĩa kiểu dữ liệu cấu trúc với `typedef`

- Với việc sử dụng `typedef` chúng ta không phải viết `"struct tên_cấu_trúc"` khi khai báo biến có kiểu dữ liệu cấu trúc.
- Chúng ta có thể sử dụng kiểu dữ liệu cấu trúc được định nghĩa bởi `typedef` như cách dùng với các kiểu dữ liệu chuẩn.
- Ví dụ: như chúng ta đã định nghĩa kiểu dữ liệu cấu trúc `student_t` và `car_t` ở trên thì có thể khai báo biến như sau:

```
car_t mycar;  
student_t excellentP;
```



# Truy nhập các trường dữ liệu của cấu trúc

- Cú pháp:

```
ten_bien_cau_truc.ten_truong
```

- Ví dụ:

```
struct point_2D  
{  
    char ten_diem;  
    float x, y;  
} p;
```

– Truy cập tên của 1 điểm: `p.ten_diem`

# Phép gán giữa các biến cấu trúc

- Giả sử ta có 2 biến cấu trúc là a và b có **cùng kiểu dữ liệu** là một cấu trúc nào đó.
- Phép gán cấu trúc có cú pháp tương tự như phép gán thông thường:  
`bien_cau_truc_1 = bien_cau_truc_2;`
- Cấu trúc là một kiểu dữ liệu nên cũng tồn tại dạng con trỏ và mảng như các kiểu dữ liệu chuẩn khác.

# BÀI TẬP 14.1

- Viết chương trình có khai báo kiểu dữ liệu có cấu trúc ngày tháng năm.
- Viết chương trình con cho phép nhập dữ liệu một ngày trong năm bất kỳ đúng định dạng của ngày tháng năm.
- Viết chương trình con so sánh 2 ngày 1 và ngày 2 là tham số được truyền vào:
  - Trả về -1 nếu ngày 1 trước ngày 2
  - Trả về 0 nếu ngày 1 và ngày 2 trùng nhau
  - Trả về 1 nếu ngày 1 sau ngày 2.
- Viết đoạn chương trình kiểm nghiệm việc nhập vào 2 ngày và hiển thị thông báo so sánh.

# Xóa Buffer khi đọc dữ liệu từ bàn phím

- Môi trường Windows:

- `fflush(stdin);`

- Các môi trường khác như linux: tự viết hàm như sau:

```
void clear_buffer() {  
    int ch;  
    while ( (ch=getchar()) != '\n' &&  
            ch!=EOF) ;  
}
```

# BÀI TẬP 14.2

- Lập chương trình quản lí điểm thi môn C cho một lớp học có  $n$  học sinh,  $n \leq 10$ .
- Thông tin về điểm thi môn C của từng học sinh được lưu dưới dạng bản ghi:
  - Tên học sinh: không vượt quá 20 kí tự
  - Mã học sinh: số nguyên
  - Điểm tổng kết: số thực
- Viết chương trình nhập dữ liệu cho một lớp có  $n$  học sinh, với  $n$  là số nguyên được nhập từ bàn phím và  $n \leq 10$ .
- 1/ Sắp xếp danh sách lớp theo thứ tự điểm thi giảm dần, 2/ thứ tự tên theo ABC và đưa ra màn hình theo dạng:

STT	MaHS	Ho va Ten	Diem
1	003	Nguyen van X	8.2

# BÀI TẬP 14.3

- Lập chương trình quản lí bán hàng với các yêu cầu như sau: Thông tin về hàng hóa được lưu dưới dạng bản ghi:
  - Tên hàng: không vượt quá 20 kí tự
  - Mã hàng: 6 ký tự
  - Đơn giá: số thực
  - Số lượng bán: số nguyên
- Viết chương trình nhập dữ liệu cho 1 ngày bán. Việc nhập sẽ dừng lại khi tên hàng nhập là \*\*\*
- Sắp xếp lại danh sách đã nhập theo Mã hàng.
- Đưa ra màn hình danh sách các mặt hàng có tổng giá trị bán > 500000 đồng theo dạng sau:

+-----+-----+-----+-----+-----+-----+					
STT	Ma hang	Ten hang hoa	Don gia	So luong	Tong tien
+-----+-----+-----+-----+-----+-----+					
01	TL02	Tu lanh	40000	20	4000000
. . . . .					
-----					
TONG CONG: 12300000 VNĐ					

# BÀI TẬP 14.4

Lập chương trình ghép phách bài thi có các yêu cầu như sau:

- Thông tin về thí sinh gồm có:
  - Tên học sinh: không vượt quá 20 kí tự
  - Mã học sinh: không quá 8 ký tự
  - Số phách: số nguyên.
- Điểm bài thi gồm có:
  - Số phách: số nguyên.
  - Điểm: số thực

Có  $n$  thí sinh tham gia thi,  $n \leq 10$  và nhập từ bàn phím.

Lập chương trình thực hiện các công việc sau:

a/ Nhập danh sách thí sinh. Sắp xếp thí sinh theo thứ tự tăng dần của số phách.

b/ Nhập danh sách kết quả bài thi. Sắp xếp danh sách đó theo thứ tự tăng dần của số phách.

c/ Sử dụng kết quả của câu a) và b) ở trên để đưa ra màn hình kết quả thi dưới dạng:

+-----+-----+-----+-----+			
STT	MaThiSinh	Ho va Ten	Diem
+-----+-----+-----+-----+			
1	003	Nguyen van X	8.2