

25  
SOICT

YEARS ANNIVERSARY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# Tuần 14. Cấu trúc - struct

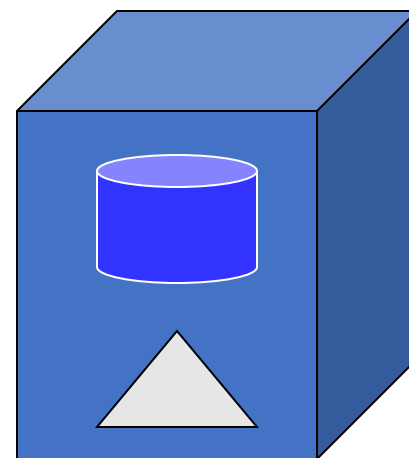
## Kiểu dữ liệu tự định nghĩa

# Chủ đề của tuần này

- Nội dung
  - Ôn lại bài trên lớp
    - Khai báo dữ liệu có cấu trúc
    - Sử dụng typedef
    - Truy nhập các biến trong cấu trúc.
  - Làm các bài tập lập trình

# Dữ liệu có cấu trúc

- Một cấu trúc trong C là tập hợp của các phần tử có kiểu khác nhau.
- Các cấu trúc thường hữu ích trong việc tạo ra các cấu trúc dữ liệu lớn và phức tạp hơn so với các kiểu dữ liệu cũ.



# Định nghĩa/ khai báo một cấu trúc ( **struct** ) trong C

```
struct tên-cấu-trúc  
{  
    field-type1 field-name1;  
    field-type2 field-name2;  
    field-type3 field-name3;  
    . . .  
};
```

# Ví dụ

- Ta có thể định nghĩa một kiểu để biểu diễn thông tin sinh viên với tên, tuổi và điểm như sau:

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

# Ví dụ

- Tạo một kiểu mới cho ô tô:
  - Ô tô phải có hãng sản xuất, mô đun ( kiểu string) và năm sản xuất ( kiểu số nguyên)

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

# Khai báo và khởi tạo biến

- Phải dùng từ khóa **struct** trong khai báo

**struct** student s1;

**struct** car mycar;

**struct** student s1 = (“Nguyen Le”, 19, 8.0);

**struct** car mycar = (“Fiat”, "Punto", 2004);



# Khai báo cấu trúc với typedef

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

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

Bây giờ chương trình có thêm kiểu dữ liệu mới- **student\_t** và **car\_t**

# Khai báo biến

- Bằng cách dùng **typedef**, chúng ta không phải viết “**struct student**” mỗi khi sử dụng cấu trúc nữa!
- Ví dụ khai báo sau:

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

# Truy cập các phần tử trong cấu trúc

- Sử dụng dấu chấm giữa tên của cấu trúc và tên trường.

```
car_t mycar;
```

```
mycar.year = 2004;
```

```
student_t excellentp;
```

```
excellentp.age = 18;
```

```
excellentp.grade = 7.8;
```

# Exercises 14.1

- a) Tạo cấu trúc tên là **Date** để lưu các thông tin. Mỗi **date** có trường thông tin **day**, **month** và **year**.
- b) Viết một hàm nhập vào dữ liệu kiểu trên. Nhớ kiểm tra tính chính xác của dữ liệu.
- c) Viết hàm **datecmp** để so sánh 2 date. Hàm này trả về
  - 1 nếu date thứ nhất là trước date thứ hai
  - 0 nếu 2 date là trùng nhau.
  - 1 nếu date thứ nhất là sau date thứ hai
- d) Viết chương trình yêu cầu người dùng nhập vào 2 date và in ra kết quả so sánh.

*Ví dụ: 2/10/1997 là sau 23/8/1997*

# Solution: Structure Declaration

```
#include <stdio.h>
typedef struct date
{
    unsigned char day;
    unsigned char month;
    int nam;
}date_t;
```

# Solution: Input function

```
date input_date() {
    date tmp;
    do {
        printf("The day (between 1 and 31):");
        scanf("%u", &tmp.day);
    } while ((tmp.day <1) || (tmp.day >31))
    do {
        printf("The month (between 1 and 12):");
        scanf("%u", &tmp.month);
    } while ((tmp.day <1) || (tmp.day >12))
    do {
        printf("The year (between 1 and 10000):");
        scanf("%d", &tmp.year);
    } while ((tmp.day <1) || (tmp.day >1000))

    return tmp;
}
```

# Solution: Date compare function

```
int datecmp(date d1, date d2) {  
    if (d1.year < d2.year) return -1;  
    else if (d1.year > d2.year) return 1;  
    else {  
        if (d1.month < d2.month) return -1;  
        else if (d1.month > d2.month) return 1;  
        else {  
            if (d1.day < d2.day) return -1;  
            else if (d1.day > d2.day) return 1;  
            else return 0;  
        }  
    }  
}
```

# Solution: main program

```
int main() {
    date date1, date2;
    int m;
    printf("Enter the first date.\n");
    date1= input_date();
    printf("Enter the second date.\n");
    date2= input_date();
    m = datecmp(date1, date2);
    if (m==0) printf("Two date are identical.\n");
    else if (m<0) printf("%d/%d/%d is before %d/%d/%d\n",
date1.day, date1.month, date1.year, date2.day, date2.month,
date2.year);
    else printf("%d/%d/%d is after %d/%d/%d\n", date1.day,
date1.month, date1.year, date2.day, date2.month, date2.year);
    return 0;
}
```



# Exercise 14.2

- Viết chương trình sử dụng cấu trúc để lưu thông tin thời tiết cho một tháng:
  - *Total Rainfall* // Tổng lượng mưa
  - *High Temperature* // Nhiệt độ cao nhất
  - *Low Temperature* // Nhiệt độ thấp nhất
  - *Average Temperature* // Nhiệt độ trung bình
- Chương trình nên có một mảng 12 phần tử kiểu cấu trúc để lưu dữ liệu thời tiết cho cả năm. Khi chương trình chạy, nó yêu cầu người dùng nhập vào dữ liệu cho từng tháng và sau đó tính toán và hiển thị lượng mưa trung bình, tổng lượng mưa trong năm, nhiệt độ cao nhất và thấp nhất trong năm.
- Kiểm tra dữ liệu nhập vào: Chỉ chấp nhận nhiệt độ trong khoảng -40 và 50 độ Celcius.

# solution

```
#include <stdio.h>
typedef struct wt{
    int total_rain; // in mm
    int high_temp; // in celcius
    int low_temp;
}weather;
const int MONTHS = 12;
```

# solution

```
int main() {
    weather data[MONTHS];
    int i;
    char months[MONTHS][10]={"January", "February", "March",
    "April"};
    int total_rain = 0;
    int max_temp = -100;
    int max_month = 0;
    int min_temp = 100;
    int min_month = 0;
    // read input data for each month
    for (i = 0; i < MONTHS; i++) {
        printf("Weather Data for %d: \n", months[i]);
        printf("    Total Rainfall (mm): ");
        scanf("%d",&data[i].total_rain);
        printf("    High Temperature (C): ");
        scanf("%d",&data[i].high_temp);
        printf("    Low Temperature (C): ");
        scanf("%d",&data[i].low_temp);
    }
```

# solution

```
// find min, max temperature
for (i = 0; i < MONTHS; i++) {
    total_rain += data[i].total_rain;
    if (max_temp < data[i].high_temp) {
        max_temp = data[i].high_temp;
        max_month = i;
    }
    if (min_temp > data[i].low_temp) {
        min_temp = data[i].low_temp;
        min_month = i;
    }
}

printf ("\nWeather Statistic for the Year:\n");
printf ("  Total Rainfall: %d mm\n", total_rain);
printf (" Monthly Rainfall: %2.3f mm\n",
double(total_rain)/MONTHS );
printf (" Highest Temperature: " : %d °C in %s \n",
max_temp, months[max_month]);
printf (" Lowest Temperature: " : %d °C in %s \n",
min_temp, months[min_month]);

return 0;
```

# Exercise 14.3

- Viết chương trình quản lý sinh viên sử dụng cấu trúc:

```
typedef struct
{
    char id[6];
    char name[31];
    float grade;
    char classement
} student;
```

Sinh viên được xếp hạng dựa trên điểm số theo tiêu chí sau :

- từ 9 tới 10: A (Excellent)
- từ 8 tới 9: B (Good)
- từ 6.5 tới 8: C (Medium)
- < 6.5 : D (Bad)

# Exercise 14.3

- Chương trình nên đọc dữ liệu từ bàn phím cho  $n$  sinh viên, sau đó in ra danh sách sinh viên theo thứ tự xếp hạng giảm dần như sau:

Họ tên	Điểm	Hạng
Dao Tiem	9.3	A
Dinh Lan	8.2	B
Bui Luu Van	5.7	D

# Solution:

```
#include <stdio.h>
#include <string.h>
#define MAX 100
typedef struct {
    char id[6];
    char name[31];
    float grade;
    char classement
} student;

void printStudent(student s)
{
    printf("ID | Name | Grade | Classment\n");
    printf("%s | %s | %1.1f | %c\n", s.id, s.name,
        s.grade, s.classement);
}
```

# Solution:

```
int main() {
    int i, n;
    student std_list[MAX], tmp;
    printf("Enter the number of student (>0):");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("ID:"); gets(std[i].id);
        printf("name:"); gets(std[i].name);
        printf("Grade:"); scanf("%f",&std[i].grade);
        if (std[i].grade >= 9 && std[i].grade <= 10)
            std[i].classment = 'A';
        else if (std[i].grade >= 8 && std[i].grade < 9)
            std[i].classment = 'B';
        else if (std[i].grade >= 6.5 && std[i].grade < 8)
            std[i].classment = 'C';
        else std[i].classment = 'D';
    }
}
```



# Solution: Sort the student list

```
for(i=0; i<n-1; i++)
    for (j=i+1; j<n; j++)
        if (std[i].grade < std[j].grade){
            tmp=std[i];
            std[i]=std[j];
            std[j]=tmp;
        }
for(i=0; i<n; i++) printStudent(std[i]);
return 0;
}
```

# Exercise 14.4

- Xây dựng kiểu dữ liệu mới để biểu diễn phân số (fraction). Sử dụng kiểu dữ liệu này để viết chương trình tính toán phân số. Chương trình cần có các chức năng sau.
  - Nhập dữ liệu cho một mảng các phân số
  - In nội dung của mảng phân số
  - Nghịch đảo tất cả phân số trong mảng
  - So sánh hai phân số

# Solution:

```
#include <stdio.h>
#define MAX 100
typedef struct {
    int numerator;
    int Denominator;
}fraction;

void fractionInput(fraction *ps);
void fractionOutput (fraction ps);
void fractionArrayInput(fraction dsps[], int n);
void fractionArrayOutput(fraction dsps[], int n);
int fractionCmp(fraction ps1, fraction ps2);
fraction inverse(fraction ps);
void inverseArray(fraction dsps[], int n);
```

# Solution:

```
void fractionInput(fraction *ps){
    int n,d;
    printf("Numerator:"); scanf("%d",&n);
    ps->numerator = n;
    do {
        printf("Denominator:"); scanf("%d",&d);
    } while (d==0);
    ps->denominator = d;
}

void fractionArrayInput(fraction dsps[], int n){
    int i;
    for (i=0; i<n; i++){
        printf("Data input for the i-th fraction:\n");
        fractionInput(dsps[i]);
    }
}
```

# Solution:

```
void fractionOutput(fraction ps){
    printf(" %d/%d ", ps.numerator,
ps.denominator);
}

void fractionArrayOutput(fraction dsps[], int n){
    int i;

    printf("Data output for the array of
fraction:\n");

    for (i=0; i<n; i++){
        fractionOutput(dsps[i]);
    }

    printf("\n");
}
```

# Solution:

```
fraction inverse(fraction ps){
    fraction tmp;
    if (ps.numerator ==0) {
        printf("Can not have an inverse fraction of 0!\n");
        exit(1);
    }
    tmp.numerator = ps.denominator; tmp.denominator=ps.numerator;
    return tmp;
}

int fractionCmp(fraction ps1, fraction ps2){
    long smd =(ps1.numerator*ps2.denominator
    -ps2.numerator*ps1.denominator
    )/ps1.denominator*ps2.denominator;
    if(smd> 0)return 1;
    else if (smd<0) return -1;
    else return 0;
}
```

# Solution:

```
void inverseArray(fraction dsps[], int n){
    int i;
    for(i=0; i<n; i++) {
        dsps[i]= inverse(dsps[i]);
    }
}

int main(){
    int n;
    fraction a[MAX], max, s, p;
    printf("Enter the number of fractions:");
    scanf("%d",&n);
    fractionArrayInput(a, n);
    fractionArrayOutput(a, n);
    printf("Inverse all fractions in the array.\n");
    inverseArray(a,n);
}
```

# Solution:

```
printf("Enter first fraction to compare:");
fractionInput(s);
printf("Enter second fraction to compare:");
fractionInput(p);
if (fractionCmp(s,p) ==0)
    printf("Two fractions are equals.\n");
else if (fractionCmp(s,p) < 0){
    fractionOutput(s);
    printf("is smaller than");
    fractionOutput(p);
    printf("\n");
}
return 0;
}
```



# Solution:

```
s=TongCacPS(a, n);  
printf("\nTong gia tri cac phan so co trong mang: ");  
XuatPS(s);  
  
p=TichCacPS(a, n);  
printf("\nTich gia tri cac phan so co trong mang: ");  
XuatPS(p);  
  
NghichDaoCacPS(a, n);  
printf("\nMang phan so sau khi nghich dao cac phan tu: ");  
XuatMangPS(a, n);  
return 0;  
}
```

# Exercise 14.5

- Phát triển thêm các chức năng hữu ích khác với phân số:
  - Chuyển đổi phân số thành phân số giản đơn. Ví dụ  $\frac{14}{3} = 4\frac{2}{3}$
  - Nhân, cộng hai phân số
- Tích hợp các chức năng này vào bài tập 14.4



25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)





ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG