

# **Chương 1. Cấu trúc chung của chương trình C**

I. Giới thiệu về ngôn ngữ C

II. Các phần tử cơ bản của ngôn ngữ C

III. Cấu trúc chung của một chương trình C (viết trên DOS)

IV. Các bước viết và chạy thử chương trình C

1

## **I. Giới thiệu về ngôn ngữ C**

1. Ngôn ngữ lập trình C

2. Trình biên dịch C

2

# 1. Ngôn ngữ lập trình C

- w Năm 1973 ngôn ngữ lập trình C ra đời với mục đích ban đầu là để viết hệ điều hành Unix trên máy tính mini PDP. Sau đó C đã được sử dụng rộng rãi trên nhiều loại máy tính khác nhau và đã trở thành một ngôn ngữ lập trình có cấu trúc rất được ưa chuộng.
- w C là ngôn ngữ lập trình bậc trung, có tính cấu trúc và định kiểu.

3

# 2. Trình biên dịch C

- w Borland C++ 3.1 là một chương trình biên dịch các chương trình C++ và C viết trên DOS và cả trên Windows.
- w Microsoft Visual C++ 6.0
- w **Borland C++ 5.5 command-line: Chỉ là trình biên dịch, đây là phần mềm miễn phí.**
- w Trên Linux, g++ hoặc gcc

4

## II. Các phần tử cơ bản của ngôn ngữ C

1. Bộ ký tự
2. Từ khoá
3. Các tên tự đặt
4. Các tên chuẩn
5. Dấu chấm phẩy
6. Lời chú thích

5

### 1. Bộ ký tự của ngôn ngữ C

**w** Mọi ngôn ngữ lập trình đều được xây dựng trên một bộ ký tự nào đó. Các ký tự ghép lại với nhau tạo thành các từ. Các từ lại được liệt kết với nhau theo một quy tắc nào đó để tạo thành các câu lệnh. Một chương trình bao gồm nhiều câu lệnh diễn đạt một thuật toán để giải một bài toán nào đó.

**w** Bộ ký tự của ngôn ngữ C gồm có các ký tự sau:

- n 26 chữ cái hoa: A, B, C, ..., Z và 26 chữ cái thường: a...z
- n 10 chữ số: 0, 1, 2, ..., 9
- n Các ký hiệu toán học: + - \* / = ) (

6

# 1. Bộ ký tự của ngôn ngữ C

- n Ký tự gạch nối \_
- n Các dấu chấm câu và các ký tự đặc biệt khác: . , ; : [] ? ! \ & | % # \$ ....
- n Dấu cách là một khoảng trống dùng để ngăn cách giữa các từ.

**Chú ý:** Khi viết chương trình ta không được sử dụng các ký tự không có trong tập ký tự trên.

7

# 2. Từ khoá

**w** Từ khoá là những từ của riêng ngôn ngữ C. Chúng được sử dụng cho các kiểu dữ liệu, toán tử và câu lệnh.

**w** Các từ khoá của C gồm có:

asm	_asm	__asm	auto	break	case
cdecl	_cdecl	__cdecl	char	class	const
continue	_cs	__cs	default	delete	do
double	_ds	__ds	else	enum	_es
__es	_export	__export	extern	far	_far

8

## 2. Từ khoá

`__far`   `_fastcall`   `__fastcall`   `float`   `for`   `friend`  
`goto`   `huge`   `_huge`   `__huge`   `if`   `inline`  
`int`   `interrupt`   `_interrupt`   `__interrupt`   `_loadds`   `__loadds`  
`long`   `near`   `_near`   `__near`   `new`   `operator`  
`pascal`   `_pascal`   `__pascal`   `private`   `protected`   `public`  
`register`   `return`   `_saveregs`   `__saveregs`   `_seg`   `__seg`  
`short`   `signed`   `sizeof`   `_ss`   `__ss`   `static`  
`struct`   `switch`   `template`   `this`   `typedef`   `union`  
`unsigned`   `virtual`   `void`   `volatile`   `while`

9

## 3. Các tên tự đặt

- w** Tên dùng để xác định các đại lượng khác nhau trong chương trình như tên hằng, tên biến, tên hàm, tên con trỏ, tên cấu trúc, tên tệp, tên nhãn,...
- w** Tên là một dãy ký tự có thể là chữ cái, chữ số hoặc dấu gạch nối song ký tự đầu tiên phải là chữ cái hoặc dấu gạch nối. Tên không được đặt trùng với từ khoá.
- w** Một số ví dụ về tên đặt sai:
  - `3XYZ_7`   `R#3`
  - `F(x)`   `case`
  - `Al pha`

10

## 4. Tên chuẩn

**w** Tên chuẩn là các tên đã được trình biên dịch đặt. Tên chuẩn có thể là tên hằng, tên các hàm.

**Ghi nhớ:** + Các từ khoá, tên tự đặt, tên chuẩn phân biệt chữ hoa chữ thường, nghĩa là viết hoa, viết thường là khác nhau.

*Ví dụ: Tên AB khác với tên ab*

+ Riêng từ khoá, tên chuẩn luôn luôn dùng chữ thường, tên chuẩn là hằng thường là chữ hoa.

11

## 5. Dấu chấm phẩy

**w** Dấu chấm được dùng để ngăn cách giữa các câu lệnh. Dấu chấm phẩy thường đặt ở cuối câu lệnh và không thể thiếu được trong chương trình C.

*Ví dụ:*

float x;

x = 10.5;

x = 2\*x - 2.5;

12

## 6. Lời giải thích

**W** Lời giải thích do người lập trình đưa vào để cho chương trình dễ hiểu, dễ đọc. Lời giải thích có thể đặt bất kỳ đâu trong chương trình nhưng phải đặt trong cặp

```
/*          */
```

hoặc đặt sau //

**W** Dùng `/*` và `*/` khi lời giải thích nằm trên nhiều dòng, dùng `//` khi lời giải thích nằm trên một dòng.

13

### III. Cấu trúc chung của một chương trình C

```
//Khai báo sử dụng thư viện chương trình con, thư viện lớp
```

```
#include<iostream>
```

```
#include<stdio.h>
```

← Tương đương với  
USES trong PASCAL

```
using namespace std;
```

```
.....
```

```
//Khai báo các hàm (chương trình con)
```

```
.....
```

```
int main()
```

```
{
```

```
    //Khai báo các biến, hằng
```

```
    .....
```

```
    //Các lệnh của chương trình
```

```
    .....
```

```
    cout<<endl;
```

```
    return 0;
```

```
}
```

```
//Định nghĩa các hàm
```

```
.....
```

Tương đương với  
BEGIN trong PASCAL

} Thân chương trình  
chính

← Tương đương với  
END trong PASCAL

14

## IV-Các bước viết và chạy thử chương trình C

### **Bước 1:** Soạn thảo chương trình

- Chạy trình soạn thảo văn bản text (ASCII): Notepad++ (DOS) hoặc mcedit (Linux)
- Gõ vào chương trình và ghi thành tệp có đuôi .cpp

### **Bước 2:** Biên dịch chương trình

- Chạy trình biên dịch C/C++ để biên dịch chương trình: bcc32 (DOS) hoặc g++ (Linux)
- Nếu có lỗi về mặt cú pháp thì quay lại bước 1 để sửa.

### **Bước 3:** Chạy thử chương trình

- Từ hệ điều hành cho chạy chương trình
- Nhập vào dữ liệu mẫu, nếu thấy kết quả sai thì kiểm tra lại thuật giải rồi quay lại bước 1 viết lại chương trình.