

KỸ THUẬT LẬP TRÌNH

GV: NGUYỄN VĂN DANH

Thời lượng : 90 tiết

(30 Lý thuyết + 60 Thực hành)

- **Chương 1: Các khái niệm cơ bản của ngôn ngữ lập trình C/C++**
- **Chương 2: Các hàm nhập xuất**
- **Chương 3: Cấu trúc điều khiển.**
- **Chương 4: Hàm**
- **Chương 5: Mảng và chuỗi**
- **Chương 6: Con trỏ**
- **Chương 7: Kiểu dữ liệu do người dùng định nghĩa**

CÁCH TÍNH ĐIỂM

- 2 cột thường xuyên.
- 1 cột giữa kỳ

Thi cuối kỳ: Hình thức thi tự luận/ trên máy

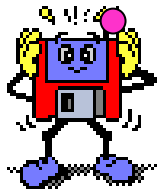
- Cài ứng dụng trên điện thoại:
 - ❖ Cài ứng dụng google classroom
 - ❖ Tham gia vào lớp có mã lớp: **x08tagi**
- Cung cấp các tài liệu để đọc thêm.

Mở đầu



➤ Tại sao lại học Kỹ thuật lập trình???

Làm thế nào để
máy tính hiểu và
trực tiếp thực hiện
được thuật toán?



Cần diễn tả thuật toán bằng một ngôn ngữ
mà máy tính hiểu và thực hiện được.

Chương trình



Ngôn ngữ để viết chương trình gọi là ngôn ngữ lập trình.

CHƯƠNG I

CÁC KHÁI NIỆM CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH C/C++

BÀI 1 : Khái quát về ngôn ngữ lập trình C/C++

BÀI 2 : Các kiểu dữ liệu cơ sở

BÀI 3 : Biểu thức – Toán tử - Câu lệnh

KHÁI QUÁT VỀ NGÔN NGỮ C/C++

Do B.W.Kerninghan & D.M.Ritchie (1970) phát triển tại Bell lab.

Tại sao lại sử dụng C?

- NNLT vạn năng.

- Tính khả chuyển (portability).

- Độ thích nghi cao, tính mềm dẻo, NNLT chuyên nghiệp.

- Tính hiệu quả.

1. Cấu trúc chung của một chương trình C

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

} Phần dẫn hướng biên dịch

} Phần khai báo

```
/* Chương trình chính */
void main()
{
    printf("Hello, world!\n");
    _getch();
}
```

} Phần chương trình chính

2. Các thành phần cơ bản của ngôn ngữ

1. Bộ chữ viết : a, b, c, ... z, A, B, C, ... , Z
 - ✓ Phân biệt chữ hoa và chữ thường
2. Tên : Tên bắt đầu bằng chữ cái hoặc dấu “_”
 - ✓ Quy tắc đặt tên : không được chứa khoảng trắng và không trùng các từ khóa
3. Từ khoá : asm, break, case, char, const, continue, if, ...

Ví dụ :

Các tên sau là sai : ho ten, int, char, 1ho, ...

Các tên sau đúng : hoten, Hoten, _ho, ...

Môi trường lập trình

- Borland C
- Borland C – for Win
- Visual C++
- C- Free
- **Dev-C++**
- Visual Studio 20...

Tải: <https://nvdanh.wordpress.com/phan-mem-mien-phi-cho-sinh-vien/>

(Anh/chị có thể search google với: nvdanh hoặc nguyen van danh, vào mục phần mềm tải và cài đặt)

BÀI 2: CÁC KIỂU DỮ LIỆU CƠ SỞ

1. Kiểu dữ liệu cơ sở

<i>Kiểu thước</i>	<i>Phạm vi</i>	<i>Kích</i>
char	0 .. 255	1 byte
int	-32768 .. 32767	2 bytes
long	-2147483648 .. 2147483647	4 bytes
unsigned	0 .. 65535	2 bytes
float	3.4e-38 .. 3.4e+38	4 bytes
double	1.7e-308 .. 1.7e+308	8 bytes

2. Khai báo hằng

- Hằng tượng trưng :

`#define <tên_hằng> <giá_trị>`

Ví dụ:

```
#define MAX 100
```

```
#define ECHO " CDKT Ly Tu Trong "
```

- Hằng biến :

`const <tên_kiểu> <tên_hằng>=<giá_trị>;`

Ví dụ :

```
const long MAX = 100L;
```

```
const char ECHO[30] = " CDKT Ly Tu Trong ";
```

3. Khai báo biến

<Kiểu_dữ_liệu> <Tên biến> [= <Giá trị>];

Giá trị của biến có thể được khởi gán khi khai báo.

Ví dụ:

int a =10;

int i, j;

float b;

float c =10.25;

char ch, ch1 = 'A';

BÀI 3 : Biểu thức – Toán tử - Câu lệnh

I. Biểu thức – Toán tử :

1. Biểu thức :

Biểu thức là một công thức tính toán bao gồm các phép toán, hằng, biến, hàm và các dấu ngoặc.

Ví dụ :

$$a + b * c$$

$$(b * b - 4 * a * c) / (2 * a)$$

$$(num \% 2) != 0$$

BÀI 3 :Biểu thức – Toán tử - Câu lệnh

<i>Toán tử</i>	<i>Thứ tự kết hợp</i>
() , [] , ->	Trái sang phải
~, ++, --, (type)	Phải sang trái
*, /, %	Trái sang phải
+, -	Trái sang phải
<<, >>	Trái sang phải
<, <=, >, >=	Trái sang phải
&	Trái sang phải
	Trái sang phải
^	Trái sang phải
&&	Trái sang phải
	Trái sang phải
==, !=, +=, -=	Phải sang trái

BÀI 3 :Biểu thức – Toán tử - Câu lệnh

II. Toán tử :

1. Toán tử so sánh :

Toán tử	Ý nghĩa	Ví dụ $a=4; b=1$
$<$	So sánh nhỏ hơn	$a < b$ kết quả false
$<=$	So sánh nhỏ hơn hay bằng	$a <= b$ kết quả false
$>$	So sánh lớn hơn	$a > b$ kết quả true
$>=$	So sánh lớn hơn hay bằng	$a >= b$ kết quả true
$==$	So sánh bằng	$a == b$ kết quả false
$!=$	So sánh không bằng	$a != b$ kết quả true

BÀI 3 :Biểu thức – Toán tử - Câu lệnh

2. Toán tử logic :

Toán tử	Ý nghĩa	Ví dụ a=true; b=false
!	Not/Phủ định/Không	!a kết quả false
&&	AND/và	a&&b kết quả false
 	OR/hoặc	a b kết quả true

BÀI 3 :Biểu thức – Toán tử - Câu lệnh

3. Toán tử Bitwise :

Khi xử lý từng bit dữ liệu, thực tế việc xử lý các thanh ghi khi lập trình phần cứng

Toán tử	Ý nghĩa	Ví dụ a=12
&	AND	$b=(a\&5)$ kết quả $b=4$
	OR	$b=(a 5)$ kết quả $b=13$
^	XOR	$b=(a\wedge 5)$ kết quả $b=9$
~	Bù 1	$b=\sim a$ kết quả $b=3$
<<	Dịch sang trái n bit	$b=(a<<2)$ kết quả $b=48$
>>	Dịch sang phải n bit	$b=(a>>2)$ kết quả $b=3$

BÀI 3 : Biểu thức - Toán tử - Câu lệnh

Ví dụ :

a=12 biểu diễn sang nhị phân 4 bit là : **1100**

c=5 biểu diễn sang nhị phân 4 bit là : **0101**

Toán tử	&		^	~	<<	>>
a=1100 C=0101	0100 (b=4)	1101 (b=13)	1001 (b=9)	0011 (b=3)	110000 (b=48)	011 (b=3)

&	0	1
0	0	0
1	0	1

	0	1
0	0	1
1	1	1

^	0	1
0	0	1
1	1	0

BÀI 3 : Biểu thức - Toán tử - Câu lệnh

4. Toán tử tăng giảm :

++ : Tăng lên 1 đơn vị.

-- : Giảm 1 đơn vị

Toán tử	Ý nghĩa	Ví dụ a=5
++	Tăng 1 đơn vị	++a kết quả a=6
--	Giảm 1 đơn vị	--a kết quả a=4

<u>Ví dụ :</u> Tiền tố	<u>Ví dụ :</u> Hậu tố
<pre>int a=5; printf("a=%d",++a);</pre> Kết quả : a = 6	<pre>int a=5; printf("a=%d",a++);</pre> Kết quả a = 5

BÀI 3 : Biểu thức - Toán tử - Câu lệnh

4. Các toán tử khác :

a. Toán tử gán :

$\langle \text{Biến} \rangle = \langle \text{Biểu_thức} \rangle;$

- Biểu thức gán:

$\langle \text{Biến} \rangle = \langle \text{Biểu_thức} \rangle$

Ví dụ:

`int x = 5;`

`int a = b = 5;`

b. Biểu thức gán rút gọn (Toán tử số học ghép) :

$\langle \text{Biến} \rangle = \langle \text{Biến} \rangle \langle \text{op} \rangle \langle \text{Biểu_thức} \rangle$

$\langle \text{Biến} \rangle \langle \text{op} \rangle = \langle \text{Biểu_thức} \rangle$

Ví dụ:

`i = i + exp;      -->`

`i += exp;`

BÀI 3 : Biểu thức - Toán tử - Câu lệnh

c. Phép toán lấy địa chỉ biến :

&<Biến>

Ví dụ: `int n;`
 `scanf("%d",&n);`

5. Toán tử điều kiện :

(condition? expr1 : epr2)

Condition : là biểu thức logic cho kết quả là true hoặc false

expr1 : biểu thức, hằng, biến, hàm bất kỳ tương ứng với condition là true

expr2 : biểu thức, hằng, biến, hàm bất kỳ tương ứng với condition là false

Ví dụ:

`max = (a>b) ? a : b;`

BÀI 3 : Biểu thức - Toán tử - Câu lệnh

II. Lệnh và khối lệnh :

- *Trong C/C++ mỗi lệnh đơn là một khai báo hay phát biểu được ghi trên một dòng.*
- *Một khối lệnh là tập các lệnh đơn và được bao trong cặp ngoặc nhọn { }.*

Ví dụ :

```
int count=5;//lệnh đơn
printf("Cac so tu 5 den 0 :\n");//lệnh đơn
while(count>=0)
{
    printf("Cac so tu 5 den 0 :\n");//lệnh đơn
    count--;//lệnh đơn
}
```

} khối lệnh

BÀI 3 : Biểu thức - Toán tử - Câu lệnh

III. Chuyển đổi kiểu tự động:

char -> int -> long -> float -> double -> long double

- Phép toán chuyển đổi kiểu:

(<Kiểu>) <Biểu_thức>

Ví dụ:

int n = (int) 10.24; kết quả là 10

CÂU HỎI VÀ BÀI TẬP

[1] Kết quả của phép tính:

a. $43\%4$

b. $201/4$

c. $5+2*3$

d. $37/4/3$

e. $37\%5\%2$

f. $(2>4 / 2) \parallel !(49.25 + 2 < 50)$

g. $(3 > 9\%7) \&\& ('a' == 97)$

h. $(7/5 == 7.0/5) \parallel (7/5.0 == 7.0/5)$

[2] Nếu x , a và b là các biến kiểu `int`, thì giá trị của chúng là bao nhiêu khi thực hiện các câu lệnh sau:

$x = 30; a = x++; b = ++x;$

[3] Giả sử trước lúc thực thi: $m = 6; n = 4$. Cho biết trị của m và n sau khi thực thi mỗi câu lệnh sau:

a) $m *= n++;$

b) $m += --n;$

[4] Cho biết kết quả khi thực hiện chương trình sau:

```
#include <iostream.h>  
int x = 4; // Khai báo và gán 4 cho biến x  
main()  
{  
    cout<< x++ << '\n'; // xuất giá trị của x++  
    cout<< x; // xuất giá trị của x  
}
```

CÂU HỎI VÀ BÀI TẬP

[5] Câu lệnh sau đúng không?

a) `int 1a ;`

b) `int a,b = 7;`

`a = ++b += 2;`

[6] Cho biết giá trị của các biến x, y, z sau khi thực hiện các câu lệnh sau và ý nghĩa của từng lệnh:

`int x, y, z;`

`x = y = z = 5;`

`x *= y += z -= 4;`

CÂU HỎI VÀ BÀI TẬP

[7] Biểu thức sau được lượng giá như thế nào?

$(x < y ? -1 : (x == y ? 0 : 1));$

[8] Viết chương trình nhập vào từ bàn phím thời gian của một công việc nào đó là N giây. Hãy chuyển đổi và viết ra trên màn hình số thời gian trên dưới dạng số giờ, số phút và số giây mà công việc thực hiện.

Ví dụ:

Nhập số giây: 36125 ↵

N = 10 giờ, 2 phút, 5 giây

CÂU HỎI VÀ BÀI TẬP

[9] Viết chương trình nhập vào số nguyên dương N có hai chữ số từ bàn phím, xuất ra màn hình tổng các chữ số của N .

Ví dụ:

Nhập N có 2 chữ số: 45 ↵

Tổng của các chữ số của 45 là 9.

Hướng dẫn: Sử dụng

➤ phép toán / và %

CÂU HỎI VÀ BÀI TẬP

[10] Viết chương trình nhập vào một số nguyên dương N có ba chữ số từ bàn phím, xuất ra màn hình tổng các chữ số của N .

[11] Viết chương trình nhập từ bàn phím 3 số nguyên a, b, c và in ra màn hình số lớn nhất.