

Phần 2: NN lập trình C++

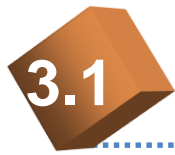


CHƯƠNG 3 CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ C++





Chương 3 – Các yếu tố cơ bản của NN C++



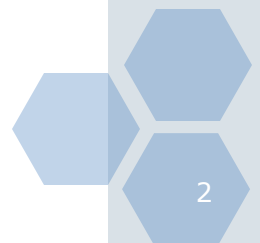
3.1 CÁC YẾU TỐ CƠ BẢN



3.2 BIỂU THỨC VÀ CÁC TOÁN TỬ



3.3 NHẬP/XUẤT DỮ LIỆU CƠ BẢN





3.1 Các yếu tố cơ bản

3.1.1 Tập các ký tự dùng trong C++ (character set)

- Các kí tự chữ hoa: A, B, C, . . . , Z
- Các kí tự chữ thường: a, b, c, . . . , z
- Các chữ số: 0, 1, . . . , 9
- Các kí tự dấu: , . ! ? : . . .
- Các kí tự trắng: ENTER, BACKSPACE, khoảng trắng.
- Các kí tự đặc biệt khác: + - * / ^ | # \$ & % () [] _ = ~ ' "
- ..

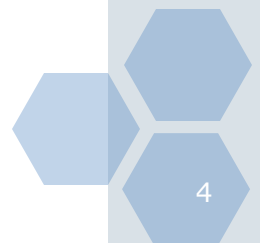




3.1 Các yếu tố cơ bản

3.1.2 Từ khóa (key word)

- Là từ dành riêng và có ngữ nghĩa xác định do NNLT quy định. Mọi NNLT đều có một bộ từ khóa riêng
- Từ khóa trong C++ bao gồm:
 - Tên kiểu dữ liệu
 - Tên toán tử
 - Tên câu lệnh





3.1 Các yếu tố cơ bản

C++ Keywords

*Keywords common to the
C and C++ programming
languages*

<code>auto</code>	<code>break</code>	<code>case</code>	<code>char</code>	<code>const</code>
<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>
<code>enum</code>	<code>extern</code>	<code>float</code>	<code>for</code>	<code>goto</code>
<code>if</code>	<code>int</code>	<code>long</code>	<code>register</code>	<code>return</code>
<code>short</code>	<code>signed</code>	<code>sizeof</code>	<code>static</code>	<code>struct</code>
<code>switch</code>	<code>typedef</code>	<code>union</code>	<code>unsigned</code>	<code>void</code>
<code>volatile</code>	<code>while</code>			

C++ only keywords

<code>asm</code>	<code>bool</code>	<code>catch</code>	<code>class</code>	<code>const_cast</code>
<code>delete</code>	<code>dynamic_cast</code>	<code>explicit</code>	<code>false</code>	<code>friend</code>
<code>inline</code>	<code>mutable</code>	<code>namespace</code>	<code>new</code>	<code>operator</code>
<code>private</code>	<code>protected</code>	<code>public</code>	<code>reinterpret_cast</code>	
<code>static_cast</code>	<code>template</code>	<code>this</code>	<code>throw</code>	<code>true</code>
<code>try</code>	<code>typeid</code>	<code>typename</code>	<code>using</code>	<code>virtual</code>
<code>wchar_t</code>				



3.1 Các yếu tố cơ bản

3.1.3 Định danh (identifier)

- ❖ Dùng để đặt tên cho các đối tượng của NSD như hằng, biến, hàm, ...
- ❖ Quy tắc đặt tên trong C++
 - Tên được bắt đầu bằng một kí tự chữ cái hoặc dấu gạch nối (underscore), tiếp theo sau là dãy các kí tự chữ hoặc số hoặc dấu gạch nối, và không cho phép có khoảng trắng ở giữa
 - Mọi tên phải được khai báo trước khi sử dụng. TRÌNH BIÊN DỊCH sẽ báo lỗi undefined symbol trong quá trình biên dịch chương trình nếu vi phạm nguyên tắc này.
 - Tên không trùng với từ khóa của C++
 - "C/C++" phân biệt chữ hoa và chữ thường (Case sensitive).
 - Trong cùng một phạm vi (scope), không được đặt tên trùng nhau.

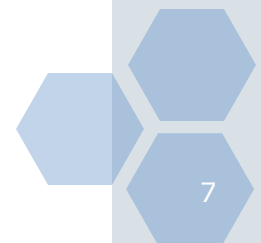


3.1 Các yếu tố cơ bản

3.1.4 Các kiểu dữ liệu cơ bản (data type)

Khái niệm:

- Kiểu dữ liệu là một tập hợp các giá trị mà một biến thuộc kiểu đó có thể nhận được, trên đó xây dựng một số phép toán đặc trưng cho các kiểu dữ liệu đó.
- Khi một biến được khai báo thuộc kiểu dữ liệu nào thì máy sẽ dành cho biến đó một dung lượng thích hợp trong bộ nhớ để có thể lưu trữ các giá trị thuộc kiểu dữ liệu này





3.1 Các yếu tố cơ bản

Các kiểu dữ liệu mở rộng

Để mở rộng các kiểu dữ liệu cơ sở, “C/C++” đưa thêm các tiền tố (prefix): ***short, long, unsigned, signed*** vào trước tên các kiểu cơ sở như sau:

Tên kiểu	Kích thước
unsigned char	1 byte (0 .. 256)
Char	1 byte (-128 .. 127)
unsigned int (unsigned)	2 bytes (0 .. 65535)
short int	2 bytes (-32768 .. 32767)
unsigned long int (unsigned long)	(0 .. 4294967295)
long int (long)	4 bytes (-2147483648 .. 2147483647)
Float	4 bytes (-3.4E -38 .. 3.4E +38)
Double	8 bytes (-1.7E -308 .. 1.7E +308)
long double	10 bytes (-3.4 E -4932 .. 1.1E+4932)



3.1 Các yếu tố cơ bản

3.1.5 Khai báo biến

- Biến - variable: Một nơi trong bộ nhớ, có thể lưu các giá trị thuộc một kiểu nào đó.
- Các biến phải được khai báo tên và kiểu trước khi sử dụng

```
int    num1;
```

```
int    num2;
```

```
int    sum;
```

- Có thể khai báo nhiều biến thuộc cùng một kiểu dữ liệu trong một dòng khai báo biến.

```
int    num1, num2, sum;
```





3.1 Các yếu tố cơ bản

Quy tắc đặt tên biến

- Chuỗi ký tự (chữ cái a..z, A..Z, chữ số 0..9, dấu gạch dưới _)
- Không được bắt đầu bằng chữ số
- Phân biệt chữ hoa chữ thường.
- Không trùng với từ khóa trong C++

Ví dụ:

- Tên biến hợp lệ: h678h_m2, _adh2, taxPayment...
- Không hợp lệ: áadàn, so chia, 2n, double ...



3.1 Các yếu tố cơ bản

Quy tắc đặt tên biến

```
cin >> num1;
```

- giả sử người dùng nhập 45

num1	45
------	----

```
cin >> num2;
```

- giả sử người dùng nhập 72

num1	45
num2	72

```
sum = num1 + num2;
```

num1	45
num2	72
sum	117



3.1 Các yếu tố cơ bản

3.1.6 Khai báo hằng

- Hằng - const: là một đại lượng có giá trị không đổi trong suốt thời gian tồn tại của nó.
- Tên hằng phải là một định danh hợp lệ và phải “được khai báo trước khi sử dụng”.
- Các ví dụ

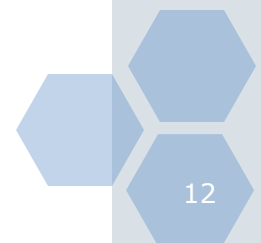
5 **số / hằng số nguyên**

5.3 **số / hằng số thực**

'Black' **Hằng chuỗi**

'C' **Hằng ký tự**

- Biến lưu giữ các giá trị hằng





3.1 Các yếu tố cơ bản

3.1.6 Khai báo hằng

- Khai báo hằng: thường được đặt trong phần khai báo toàn cục ở đầu chương trình, ngay sau các khai báo chỉ thị tiền xử lý.

- Có 2 cách khai báo hằng:

- Dùng chỉ thị tiền xử lý:

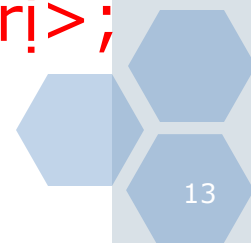
#define <tên hằng> <giá trị>

Ví dụ: #define MAX 10;

- Dùng từ khoá const:

const <tên kiểu> <tên hằng> = <giá trị>;

Ví dụ: const int MAX = 10;





3.2 Biểu thức và các toán tử

3.2.1 Biểu thức (expression)

- Là công thức tính toán bao gồm các toán hạng và các toán tử tương ứng. Các toán hạng có thể là một biến, hằng, lời gọi hàm.
- Có các loại biểu thức thông dụng sau:
 - Biểu thức gán
 - Biểu thức số học
 - Biểu thức logic.
- Trong “C/C++”, biểu thức luôn trả về một giá trị. Kiểu của biểu thức phụ thuộc vào kiểu của giá trị trả về.



3.2 Biểu thức và các toán tử

3.2.1 Biểu thức (expression)

- Chuyển đổi kiểu (type conversion): theo nguyên tắc “Kiểu có phạm vi giá trị biểu diễn nhỏ hơn sẽ được chuyển sang kiểu có phạm vi giá trị biểu diễn lớn hơn”

char* → *int* → *long* → *float* → *double* → *long double

Ví dụ:

```
int    num1, num2;  
  
float  sum;  
  
sum = num1 + num2;
```



3.2 Biểu thức và các toán tử

3.2.1 Biểu thức (expression)

- Ép kiểu (type casting): Trong một số trường hợp, ta bắt buộc phải sử dụng đến toán tử ép kiểu để tạo ra một biểu thức hợp lệ như sau:

<ten kiểu> (<biểu thức>)

Ví dụ:

```
int(8.0) % 3
```

hoặc

```
int i;
```

```
float S;
```

```
S = S + float(1/i);
```




3.2 Biểu thức và các toán tử

3.2.2 Toán tử (operator)

a. Toán tử số học

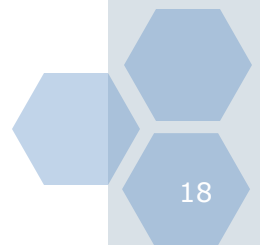
Kí hiệu	Ý nghĩa	Số ngôi	Toán hạng
+	Cộng	2	int, float, double, char
-	Trừ	2	- nt -
*	Nhân	2	- nt -
/	Chia	2	- nt -
%	Modulo	2	int, char



3.2 Biểu thức và các toán tử

b. Toán tử logic

Kí hiệu	Ý nghĩa	Số ngôi	Toán hạng
!	NOT	1	int, float, double, char
&&	AND	2	- nt -
	OR	2	- nt -

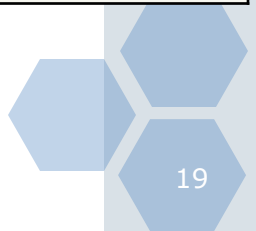




3.2 Biểu thức và các toán tử

c. Toán tử quan hệ (so sánh)

Kí hiệu	Ý nghĩa	Số ngôi	Toán hạng
<	Nhỏ hơn	2	int, float, double, char
<=	Nhỏ hơn hoặc bằng	2	- nt -
>	Lớn hơn	2	- nt -
>=	Lớn hơn hoặc bằng	2	- nt -
==	So sánh bằng	2	- nt -
!=	So sánh khác nhau	2	- nt -





3.2 Biểu thức và các toán tử

d. Toán tử gán

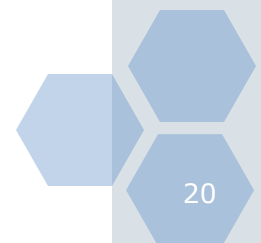
- Kí hiệu: $=$
- Cú pháp **<biến> = <biểu thức>;**
- Thực hiện: Trình biên dịch tính giá trị của <biểu thức>, sau đó gán giá trị này cho <biến>.
- Ví dụ:

$x = 1;$ // gán 1 cho biến nguyên x

$x = 2*y;$ // tính $2*y$, sau đó gán kết quả cho biến x

$z = x + 2*y;$

$a = \sin(2*b);$





3.2 Biểu thức và các toán tử

d. Toán tử gán

- C/C++ cho phép viết gọn các biểu thức gán bằng các toán tử gán:

Dạng viết thông thường	Dạng viết thu gọn	Ý nghĩa
$i = i + \langle bt \rangle$	$i += \langle bt \rangle$	Tự cộng
$i = i - \langle bt \rangle$	$i -= \langle bt \rangle$	Tự trừ
$i = i * \langle bt \rangle$	$i *= \langle bt \rangle$	Tự nhân
$i = i / \langle bt \rangle$	$i /= \langle bt \rangle$	Tự chia
$i = i \% \langle bt \rangle$	$i \% = \langle bt \rangle$	Tự module
...		



3.2 Biểu thức và các toán tử

d. Toán tử tăng (++) và giảm (--) một đơn vị

- Biểu thức có dạng

`<biến>++` `<biến>--`

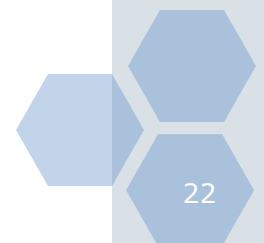
`++<biến>` `--<biến>`

- **Dạng tiền tố:** giá trị của `<biến>` được thay đổi trước khi tham gia biểu thức chung

Ví dụ: `i = 100, n = ++i`

- **Dạng hậu tố:** biểu thức chung sử dụng giá trị cũ của `<biến>`, sau đó `<biến>` mới được thay đổi giá trị.

Ví dụ: `i = 100, n = i++`





3.3 Nhập/xuất dữ liệu cơ bản

Các đối tượng nhập/xuất cơ bản

- `cin`

dòng dữ liệu vào chuẩn - Standard input stream
thường là từ bàn phím

- `cout`

dòng dữ liệu ra chuẩn - Standard output stream
thường là màn hình máy tính

- `cerr`

dòng báo lỗi chuẩn - Standard error stream hiện các
thông báo lỗi



3.3 Nhập/xuất dữ liệu cơ bản

3.3.1 Hàm cout

Cú pháp:

```
cout<<biến(hoặc hằng, hoặc biểu thức);
```

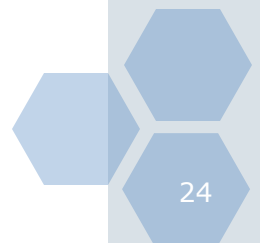
Cách sử dụng đơn giản nhất của cout là chỉ cần in ra một chuỗi:

```
cout<<"Chao mung den voi C++";
```

In ra giá trị của một biến hoặc một biểu thức:

```
int    num1 = 42, num2 = 75;
```

```
cout<<"Tong la " <<num1+num2<<endl;
```





3.3 Nhập/xuất dữ liệu cơ bản

3.3.1 Hàm cout

- Escape characters \ : In ra các ký tự đặc biệt

Chuỗi Escape	Mô tả
\n	Dòng mới. Đặt con trỏ màn hình tại đầu dòng tiếp theo.
\t	Tab. Di chuyển con trỏ đến điểm dừng tab tiếp theo.
\r	Về đầu dòng. Chuyển con trỏ màn hình tới đầu dòng hiện tại; không xuống dòng mới.
\a	Chuông. Bật chuông hệ thống.
\\	Chéo ngược. Dùng để in một dấu chéo ngược.
\"	Nháy kép. Dùng để in một dấu nháy kép.



3.3 Nhập/xuất dữ liệu cơ bản

3.3.1 Hàm cout

```
1 // Gọi chỉ thị tiền xử lý.
2 #include <iostream.h>
3
4 // hàm main để bắt đầu chương trình chính
5 void main()
6 {
7     cout << "Welcome ";
8     cout << "to C++!\n";
9
10 }
```

Nhiều dòng lệnh tạo output trên một dòng.

```
Welcome to C++!
```



3.3 Nhập/xuất dữ liệu cơ bản

3.3.1 Hàm cout

```
1 #include <iostream.h>
2
3 int main()
4 {
5     cout << "Welcome\nto\nnC++!\n";
6
7     return 0;    // đảm bảo kết quả trả về cho hàm
8 }
```

Dùng ký tự dòng mới \n để in trên nhiều dòng.

```
Welcome
to

C++!
```



3.3 Nhập/xuất dữ liệu cơ bản

3.3.2 Hàm cin

Cú pháp:

```
cin>>biến;
```

Ví dụ:

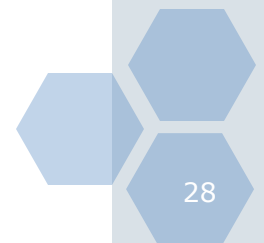
```
int n;
```

```
cin>>n;
```

Có thể nhập các biến cùng với nhau:

```
int a, b, c;
```

```
cin>>a>>b>>c;
```





3.3 Nhập/xuất dữ liệu cơ bản

3.3.2 Hàm cin

```
cin >> num1;
```

- giả sử người dùng nhập 45

num1	45
------	----

```
cin >> num2;
```

- giả sử người dùng nhập 72

num1	45
num2	72

```
sum = num1 + num2;
```

num1	45
num2	72
sum	117



3.3 Nhập/xuất dữ liệu cơ bản

3.3.3 Trình bày màn hình

Trong chương trình thường sử dụng kết hợp `cout` và `cin` để việc nhập dữ liệu được rõ ràng.

Ví dụ:

```
int n;
```

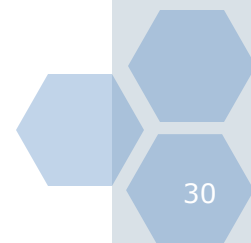
```
cout<<"Nhập vào số nguyên n:"; cin>>n;
```

Có thể nhập các biến cùng với nhau:

```
int a, b, c;
```

```
cout<<"Nhập vào 3 hệ số a, b, c";
```

```
cin>>a>>b>>c;
```





3.3 Nhập/xuất dữ liệu cơ bản

Ví dụ:

```
1  #include <iostream.h>
2
3
4  void main()
5  {
6      int num1;
7      int num2;
8      int sum;
9
10     cout << "Nhập số thứ nhất\n"
11     cin >> num1;
12
13     cout << "Nhập số thứ hai\n"
14     cin >> num2;
15
16     sum = num1 + num2;
17
18     cout << "Tổng là: " << sum << endl;
19
20 }
```

Khai báo các biến nguyên.

Nhập số thứ nhất
45
Nhập số thứ hai
72
Tổng là: 117

Nhập một số nguyên từ input chuẩn, ghi vào biến num1

endl cho kết quả là một dòng trống.

Tính toán có thể được thực hiện trong lệnh output:
Thay cho các dòng 16 và 18:
cout<<"Tổng là " << num1 + num2 << endl;



3.3 Nhập/xuất dữ liệu cơ bản

Ví dụ:

```
1  #include <iostream.h>
2
3
4  void main()
5  {
6      int num1;
7      int num2;
8      int sum;
9
10     cout << "Nhập số thứ nhất\n"
11     cin >> num1;
12
13     cout << "Nhập số thứ hai\n"
14     cin >> num2;
15
16     sum = num1 + num2;
17
18     cout << "Tổng là: " << sum << endl;
19
20 }
```

Khai báo các biến nguyên.

Nhập số thứ nhất
45
Nhập số thứ hai
72
Tổng là: 117

Nhập một số nguyên từ input chuẩn, ghi vào biến num1

endl cho kết quả là một dòng trống.

Tính toán có thể được thực hiện trong lệnh output:
Thay cho các dòng 16 và 18:
cout<<"Tổng là " << num1 + num2 << endl;

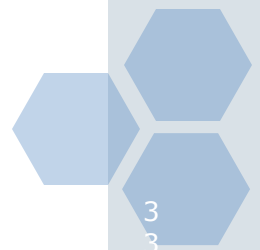


Câu hỏi củng cố bài

1. Lệnh cin trong C++ có tác dụng gì?
 - A. Là chú thích trong C++
 - B. Là lệnh khai báo một biến
 - C. Là stream đầu ra chuẩn trong C++
 - D. Là stream đầu vào chuẩn trong C++



Multiple Choice





Câu hỏi củng cố bài

2. Lệnh cout trong C++ đi kèm với cặp dấu nào?

A. Dấu <<

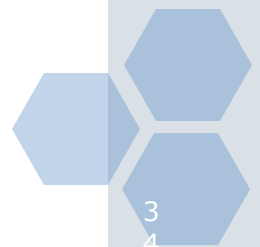
B. Dấu //

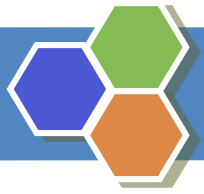
C. Dấu ||

D. Dấu >>



Multiple Choice



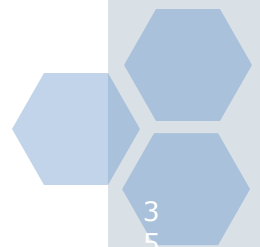


Câu hỏi củng cố bài

3. Kiểu nào dưới đây không phải là một kiểu dữ liệu trong C++?
- A. float
 - B. real
 - C. int
 - D. double



Multiple Choice

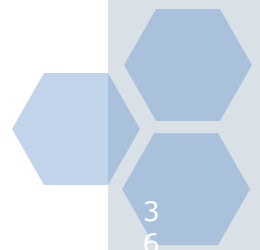




Câu hỏi củng cố bài

4. Ký hiệu nào dưới đây là toán tử logic and trong C++?
- A. |
 - B. |&
 - C. &
 - D. &&

 Multiple Choice





Câu hỏi củng cố bài

5. Toán tử nào dưới đây là toán tử so sánh bằng trong C++?

A. `:=`

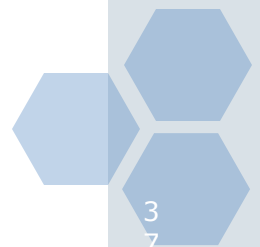
B. `=`

C. `==`

D. `!=`



Multiple Choice



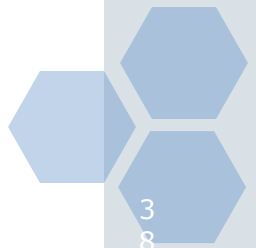


Câu hỏi củng cố bài

6. Lệnh nào sau đây định nghĩa một giá trị hằng trong C++?
- A. `const PI = 3.14.15;`
 - B. `#define PI = 3.1415`
 - C. `const float PI;`
 - D. `#define PI 3.1415`



Multiple Choice





Câu hỏi củng cố bài

7. Đoạn lệnh sau cho kết quả bao nhiêu?

A. 11

B. 10

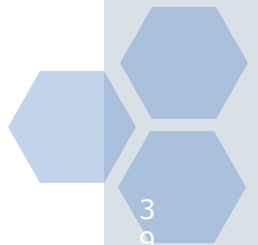
C. 6

D. 5

```
int main() {  
    int a, b;  
    a = b = 5;  
    b++;  
    cout << a++ + --b;  
    return 0;  
}
```



Multiple Choice





Câu hỏi củng cố bài

8. Đoạn lệnh sau cho kết quả bao nhiêu?

A. 10

B. 5

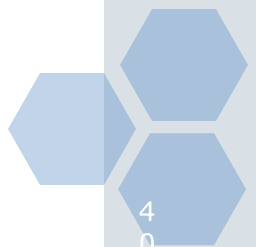
C. 6

D. 7

```
int main() {  
    int a, b;  
    a = b = 5;  
    cout << ++a;  
    return 0;  
}
```



Multiple Choice





Câu hỏi củng cố bài

9. Đoạn lệnh sau cho kết quả bao nhiêu?

A. 0

B. 1

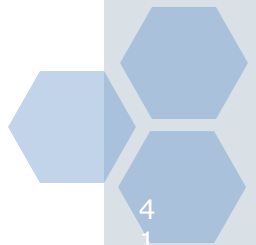
C. 3

D. Lỗi biên dịch

```
int main() {  
    int i;  
    float f = 3.14;  
    i = (int)f;  
    cout << i;  
    return 0;  
}
```

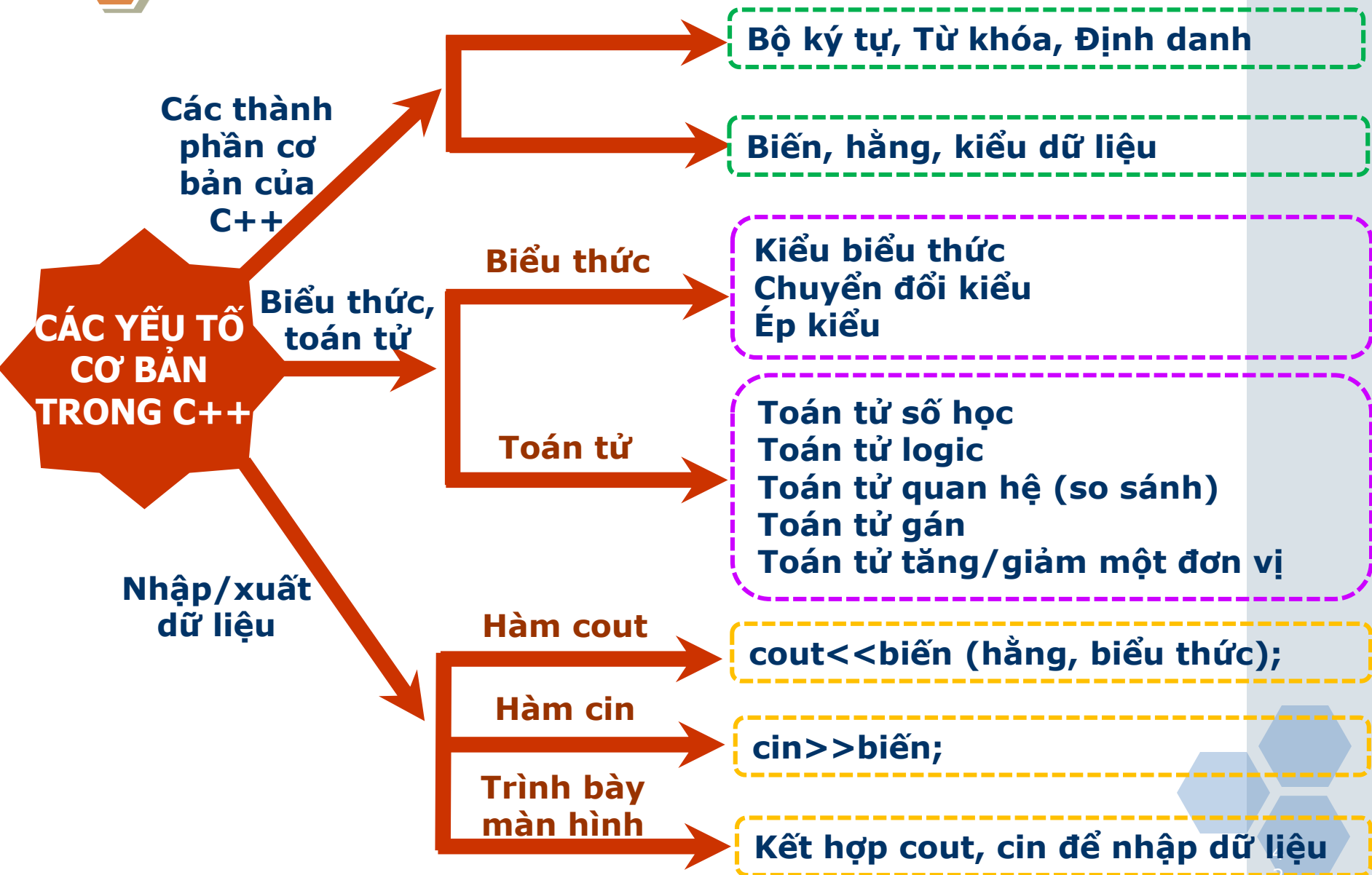


Multiple Choice





Tổng kết





Bài tập

1. Viết chương trình tính và in ra màn hình tổng hai số nguyên a , b nhập vào từ bàn phím.
2. Viết chương trình nhập vào 2 số a , b rồi đổi giá trị của biến a cho biến b và ngược lại. In kết quả sau khi đổi.
3. Viết chương trình nhập vào cạnh hình vuông a . Tính và in ra màn hình chi vi, diện tích hình vuông.

