

Chương 3

Phép toán và biểu thức

Khoa công nghệ thông tin
Trường Đại học mở tphcm

1

Mục tiêu

- Cách viết một biểu thức số học và câu lệnh gán
- Cách sử dụng một số hàm toán học trong biểu thức trong C++.
- Cách viết biểu thức luận lý để so sánh giá trị dữ liệu.
- Cách dùng cin/cout để viết câu lệnh nhập/xuất.
- Cách viết một chương trình C++ đơn giản, nhập và xử lý dữ liệu, xuất kết quả.
- Cách nhập/xuất chuỗi ký tự bằng cách dùng kiểu string trong thư viện.

2

Nội dung

1. Biểu thức
2. Biểu thức số học.
3. Biểu thức luận lý.
4. Biểu thức gán
5. Nhập/Xuất trong C++

3

3.1 Biểu thức

- Biểu thức (expression) dùng để tính toán các giá trị, được tạo thành từ toán tử (operators) và toán hạng (operands).
 - Toán tử: thực hiện tính toán với toán hạng và cho kết quả là giá trị có một kiểu.
 - Toán hạng: dữ liệu được toán tử tính toán.
- Ví dụ:
 - $2 + 3 * 5$
 - $x - y / 7$
 - $x * 10.5 + y - 16.2$
 - $2 + 3.5$
 - $6 / 4 + 3.9$
 - $5.4 * 2 - 13.6 + 18 / 2$

4

Biểu thức

- Độ ưu tiên của toán tử

Toán tử	Độ ưu tiên
! – (unary operator)	1
* / %	2
+ –	3
< <= >= >	4
== !=	5
&&	6
	7
= (assignment operator)	8

- Ví dụ:

(dài + rộng) * 2 khác với dài + rộng * 2
 $a * b + c / d$ nghĩa là $(a * b) + (c / d)$
 không phải $a * (b + c) / d$

Biểu thức

- Lưu ý khi viết biểu thức

– Thêm khoảng trắng trước và sau các toán tử trong biểu thức.

$$x1 - (-b + \sqrt{b * b - 4 * a * c}) / (2 * a);$$

để đọc hơn:

$$x1 - (-b + \sqrt{b*b - 4*a*c}) / (2*a);$$

- Không thêm khoảng trắng sau tên hàm.

\sqrt{x}

không viết

$\sqrt{ (x)}$

3.2 Biểu thức số học

- Biểu thức số gồm toán tử số học và các toán hạng là số nguyên hoặc số dấu chấm động.
 - Toán tử số học: $+$ $-$ $*$ $/$ $\%$
 - Thứ tự ưu tiên: $*$ $/$ $\%$
 $+$ $-$
 - Lưu ý: toán tử $\%$ (modulo) chỉ dùng cho kiểu số nguyên.
- Biểu thức kiểu số nguyên nếu các toán hạng có kiểu số nguyên.
- Biểu thức kiểu số dấu chấm động nếu các toán hạng có kiểu số dấu chấm động.

7

Biểu thức số học (tt)

- Ví dụ biểu thức kiểu số nguyên:


```
int dai = 4;
int rong = 5;
int dientich, chuvi;

dientich = dai * rong;
chuvi = (dai + rong) * 2
```
- Ví dụ biểu thức kiểu số dấu chấm động:


```
const double PI = 3.1414296;
double bankinh = 5.5;
double dientich;

dientich = PI * bankinh * bankinh;
```

8

Biểu thức số học (tt)

- Ví dụ viết các biểu thức số học bằng C++:

Biểu thức số học	Biểu thức C++	Ghi chú
$\frac{x + y}{2}$	<code>(x + y) / 2</code>	cần có dấu ngoặc: <code>x + y / 2</code> sẽ tính <code>x + $\frac{y}{2}$</code>
$\frac{xy}{2}$	<code>x * y / 2</code>	không cần dấu ngoặc: toán tử cùng độ ưu tiên tính từ trái sang
$\left(1 + \frac{r}{100}\right)^n$	<code>pow(1 + r / 100, n)</code>	
$\sqrt{a^2 + b^2}$	<code>sqrt(a * a + b * b)</code>	<code>a * a</code> đơn giản hơn <code>pow(a,2)</code>
$\frac{i + j + k}{3}$	<code>(i + j + k) / 3.0</code>	nếu là số nguyên thì 3.0 chuyển sang số dấu chấm động

Biểu thức số học (tt)

- Lưu ý:
 - Nếu chia hai số nguyên thì phần dư sẽ bị cắt bỏ.
`int tongsogiy = 3790;`
`int sogio = tongsogiy / 3600; // sogio == 1`
 - Phép chia lấy dư % dùng để lấy phần dư.
`int sophut = tongsogiy % 3600 / 60; // sophut == 3`
`int sogiy = tongsogiy % 3600 % 60; // sogiy == 10`
 - Tính lũy thừa và tính căn bậc 2: sử dụng hàm thư viện.
 - \sqrt{x} : `sqrt(x)`
 - x^n : `pow(x, n)`

Biểu thức số học (tt)

- Một số hàm toán học: cần `#include <cmath>`

Hàm	Mô tả	Kiểu tham số	Kiểu trả về
<code>abs(x)</code>	$ x $	int (double)	int (double)
<code>exp(x)</code>	$e^x, e = 2.718$	double	double
<code>pow(x, y)</code>	x^y	double	double
<code>sqrt(x)</code>	\sqrt{x}	double	double
<code>sin(x)</code>	$\sin x, x$: radians	double	double
<code>cos(x)</code>	$\cos x, x$: radians	double	double
<code>tan(x)</code>	$\tan x, x$: radians	double	double
<code>log(x)</code>	$\ln(x), x > 0$	double	double
<code>log10(x)</code>	$\log_{10}(x), x > 0$	double	double

11

Biểu thức số học (tt)

- Toán tử tăng và giảm
 - Toán tử tăng 1:
 - `++variable`
 - `variable++`
 - Toán tử giảm 1:
 - `--variable`
 - `variable--`
- Ví dụ:
 - `++count; hoặc count++;`
 - `--count; hoặc count--;`
 - `x = 5;`
 - `y = ++x; //x=6, y=6`
 - `x = 5;`
 - `y = x++; //y=5, x=6`

12

Biểu thức số học (tt)

- Ví dụ: nhập số thực và in ra kết quả các biểu thức

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Nhap so double: ";
    double n;
    cin >> n;
    cout << "n = " << n << endl
         << "n+1 = " << n + 1 << endl
         << "3*n = " << 3 * n << endl
         << "n^2 = " << n * n << endl
         << "n/2 = " << n / 2 << endl
         << "Can cua n = " << sqrt(n) << endl;
    return 0;
}
```

13

Biểu thức số học (tt)

- Biểu thức có thể chứa các toán hạng thuộc nhiều kiểu dữ liệu khác nhau
- Kết quả trả về của biểu thức chỉ có thể là một kiểu dữ liệu
- Có hai cách chuyển đổi kiểu:
 - Chuyển đổi kiểu ngầm định (tự động)
 - Chuyển đổi kiểu tường minh

14

Biểu thức số học (tt)

- Chuyển đổi loại ngầm định:
 - Được thực hiện bởi trình biên dịch.
 - Quy tắc: tất cả các kiểu dữ liệu của biến được nâng cấp lên kiểu dữ liệu của biến có kiểu dữ liệu lớn nhất.
- bool → char → short int → int
 → unsigned int → long → unsigned long
 → long long → float → double → long double
- Một số chuyển đổi ngầm định có thể làm mất thông tin

```
double d = 2.5;
int i = 2;
double d2 = d / i; //d2 = 1.25
int i2 = d / i; //i2 = 1
d2 = d / i; //d2 = 1.25
i2 = d / i; //i2 = 1
```

Phép toán và biểu thức

15

15

Biểu thức số học (tt)

- Ví dụ về chuyển đổi kiểu tự động:

```
#include <iostream>
using namespace std;
int main()
{
    int x = 10;
    char y = 'a';
    x = x + y;
    // x is implicitly converted to float
    float z = x + 1.0;
    cout << "x = " << x << endl //x = 107
         << "y = " << y << endl //y = 'a'
         << "z = " << z << endl; // z = 108
    return 0;
}
```

```
x = 107
y = a
z = 108
```

Phép toán và biểu thức

16

16

Biểu thức số học (tt)

- Ví dụ về chuyển đổi kiểu tự động:

- Chuyển độ Celcius sang độ Fahrenheit:
- Bị mất thông tin do kết quả là số nguyên:

```
double doC;
cin >> doC;
double doF = 9 / 5 * doC + 32; // 9/5=1, không phải 1.5
```

- Giải pháp: chuyển 9 hoặc 5 hoặc cả hai sang double

```
double doC;
cin >> doC;
double doF = 9.0 / 5 * doC + 32;
```

Biểu thức số học (tt)

- Chuyển đổi kiểu tường minh

`static_cast<type>(expression)`

- expression: biểu thức
- type: kiểu dữ liệu

```
static_cast<int>(7.9) 7
static_cast<double>(25) 25.0
static_cast<double>(5 + 3) = static_cast<double>(8) = 8.0
static_cast<double>(15) / 2 = 15.0 / 2 = 15.0 / 2.0 = 7.5
static_cast<double>(15 / 2) = static_cast<double>(7) = 7
```

- Ép kiểu:

- type(expression)
- (type)(expression)

```
int a = 5;
int b = 2;
double c = (double)a / b;
double d = double(a) / b;
```

3.3 Biểu thức luận lý

- Biểu thức luận lý có giá trị là true hoặc false, được tạo bởi toán tử luận lý và toán tử quan hệ.
 - Toán tử luận lý: `!`, `&&`, `||`
 - Toán tử quan hệ: `<` `>` `<=` `>=` `==` `!=`
 - Thứ tự ưu tiên: `!`
`<`, `<=`, `>=`, `>`
`==`, `!=`
`&&`
`||`
- Biểu thức luận lý được sử dụng làm điều kiện trong các cấu trúc điều khiển như `if`, `if...else`, `for`, `while`, `do...while`

Biểu thức luận lý (tt)

- Toán tử quan hệ dùng để so sánh các giá trị trong chương trình, cho kết quả là true hoặc false.

Toán tử quan hệ	Ví dụ	Ý nghĩa
<code>></code>	<code>x > y</code>	x lớn hơn y
<code><</code>	<code>x < y</code>	x nhỏ hơn y
<code>>=</code>	<code>x >= y</code>	x lớn hơn hoặc bằng y
<code><=</code>	<code>x <= y</code>	x nhỏ hơn hoặc bằng y
<code>==</code>	<code>x == y</code>	x bằng y
<code>!=</code>	<code>x != y</code>	x khác y

- Các giá trị so sánh phải có cùng kiểu dữ liệu.
 - Ví dụ: `8 < '5'`: không thể so sánh.

Biểu thức luận lý (tt)

- So sánh ký tự: dựa vào mã ASCII.

' ' < 'a' true

'R' > 'T' false

'+' < '*' false

'A' <= 'a' true

- So sánh chuỗi: theo thứ tự từ điển.

string s1 = "Hello";

string s2 = "Hi";

string s3 = "Air";

string s4 = "Bill";

s1 < s2 true

s3 < "An" true

s1 == "hello" false

s3 <= s4 true

Biểu thức luận lý (tt)

- Toán tử luận lý dùng để kết hợp các biểu thức luận lý.

Toán tử	Mô tả
!	not
&&	and
	or

- Ví dụ:

(gioitinh == NU) && (tuoi >= 60)

(diemGK >= 9) || (diemCK >= 9)

!(xeploai == 'D')

Biểu thức luận lý (tt)

- Bảng chân trị

exp1	exp2	exp1 && exp2
false	false	false
false	true	false
true	false	false
true	true	true

exp	!exp
false	true
true	false

exp1	exp2	exp1 exp2
false	false	false
false	true	true
true	false	true
true	true	true

Phép toán và biểu thức

23

23

Biểu thức luận lý (tt)

- Ví dụ:

```
bool found = true;
int age = 20;
double hours = 45.30;
double overTime = 15.0;
int count = 20;
char c = 'B';
```

!found	false
hours > 40.00	true
age == 20	true
!found && (age >= 18)	false
hours + overTime <= 75	true
(count >= 0) && (count <= 100)	true
('A' <= c) && (c <= 'Z')	true

Phép toán và biểu thức

24

24

3.4 Biểu thức gán

- Câu lệnh gán (assignment statement) dùng để gán giá trị cho biến trong khi chương trình đang thực hiện.

int a = 3;	a	<input type="text" value="3"/>	
a = 4;	a	<input type="text" value="4"/>	
int b = a;	a	<input type="text" value="4"/>	b <input type="text" value="4"/>
b = a + 5;	a	<input type="text" value="4"/>	b <input type="text" value="9"/>
a = a + 7;	a	<input type="text" value="11"/>	b <input type="text" value="9"/>

- Kiểu dữ liệu của biểu thức trong câu lệnh gán phải phù hợp với kiểu dữ liệu của biến.

25

Biểu thức gán (tt)

- Có thể dùng câu lệnh gán cho biến kiểu string:

string s1 = "alpha";	s1	<input type="text" value="alpha"/>	
s1 = "beta";	s1	<input type="text" value="beta"/>	
string s2 = s1;	s1	<input type="text" value="beta"/>	s2 <input type="text" value="beta"/>
s2 = s1 + "gamma";	s1	<input type="text" value="beta"/>	s2 <input type="text" value="betagamma"/>
s1 = s1 + "delta";	s1	<input type="text" value="betadelta"/>	s2 <input type="text" value="betagamma"/>

- Câu lệnh khai báo và khởi tạo (initialization): tạo một biến mới và gán cho biến giá trị ban đầu.
- Câu lệnh gán (assignment): gán cho biến đã có một giá trị mới.

26

Biểu thức gán (tt)

- Ví dụ chương trình tính giá bán tivi, tỷ lệ giảm giá là 15%.

```
#include <iostream>
using namespace std;
int main()
{
    const double tylegiam = 0.15;
    double giaban = 0.0;
    double sotiengiam = 0.0;
    double sotientra = 0.0;
    cout << "Nhap gia tivi: ";
    cin >> giaban;
    sotiengiam = giaban * tylegiam;
    sotientra = giaban - sotiengiam;
    cout << "So tien phai tra: " << sotientra << endl;
    return 0;
}
```

27

Biểu thức gán (tt)

- Các toán tử gán phức hợp (composite assignment operators) dùng để rút ngắn câu lệnh gán:

+= cộng gán
 -= trừ gán
 *= nhân gán
 /= chia gán
 %= modulo gán

- Ví dụ:

```
a += 7; //a = a + 7
b -= 9; //b = b - 9
c *= 2; //c = c * 2
counter += 1; //counter = counter + 1; ++counter; counter++;
```

28

3.5.1 Nhập trong C++

- Dùng câu lệnh nhập (input statement): đọc dữ liệu từ bàn phím.

cin >> variableName;

Ví dụ:

cin >> bankinh;

- Dùng câu lệnh gán (assignment statement).

variableName = expression;

Giá trị của biểu thức phải phù hợp với kiểu dữ liệu của biến.

Dấu = gọi là toán tử gán (assignment operator).

Ví dụ:

dientich = 3.14 * bankinh * bankinh;

29

Nhập trong C++ (tt)

- Sử dụng đối tượng cin và toán tử >>
- Toán tử >> sẽ lấy ra các ký tự từ đối tượng và chuyển vào bộ nhớ, cho đến khi gặp ký tự khoảng trắng (tab, space, enter).



keyboard → cin object → >> (extraction operator) → internal memory

30

Nhập trong C++ (tt)

- Ví dụ: nhập vào bán kính hình tròn, tính và in ra diện tích hình tròn.

Input	Processing	Output
bán kính	<i>Algorithm:</i> 1. nhập độ bán kính 2. tính diện tích = $3.14 * \text{bán kính} * \text{bán kính}$ 3. in ra diện tích	diện tích

```
const double PI = 3.14;
double bankinh;
double dientich;
cout << "Nhập bán kính hình tron: ";
cin >> bankinh;
dientich = PI * bankinh * bankinh;
cout << "Dien tích hình tron là: " << dientich << endl;
```

31

Nhập trong C++ (tt)

- Ví dụ: nhập một số và in ra bình phương của số đó.

Input	Processing	Output
x	<i>Algorithm:</i> 1. nhập giá trị vào x 2. in ra $x * x$	$x * x$

```
double x = 0.0;

cout << "Nhập một số: ";
cin >> x;

cout << "Bình phương là: " << x * x << endl;
```

32

Nhập trong C++ (tt)

- Ví dụ: nhập vào chiều cao (m) và cân nặng (kg), tính và in ra chỉ số BMI (body mass index) theo công thức:

$$bmi = \frac{\text{cân nặng (kg)}}{\text{chiều cao} \times \text{chiều cao (m)}}$$

Input	Processing	Output
chiều cao cân nặng	<i>Algorithm:</i> 1. nhập chiều cao 2. nhập cân nặng 3. tính BMI = cân nặng / (chiều cao * chiều cao) 4. in ra BMI	BMI

```
double kannang = 0.0;
double chieucao = 0.0;
double bmi = 0.0;
cout << "Nhap can nang va chieu cao: ";
cin >> kannang >> chieucao;
bmi = kannang / (chieucao * chieucao);
cout << "BMI la " << bmi << endl;
```

Phép toán và biểu thức

33

33

Nhập trong C++ (tt)

- Nhập dữ liệu kiểu string

getline(cin, str);

– str là biến có kiểu string

– đọc một chuỗi, kể cả khoảng trắng và lưu vào str

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string hoten;
    cout << "Nhap vao ho va ten: ";
    getline(cin, hoten);
    cout << "Chao ban " << hoten << endl;
    return 0;
}
```

Phép toán và biểu thức

34

34

3.5.2 Xuất trong C++

- Câu lệnh xuất (output statement): in dữ liệu ra màn hình.
`cout << expression or manipulator << ...;`
 - expression: tính giá trị của biểu thức sau đó in giá trị.
 - manipulator: định dạng dữ liệu xuất.
 - Ví dụ:


```
cout << 29 / 4 << endl;
cout << 'A' << endl;
cout << "Hello \nthere!" << endl;
cout << "4 + 8 = " << 4 + 8 << endl;
cout << "...1\n....2\n.....3\n";
hoặc
cout << "...1" << endl << "....2" << endl
<< ".....3" << endl;
```

Xuất trong C++ (tt)

- Môi trường lập trình thường xuất 6 chữ số có nghĩa (mặc định):
`cout << 12.345678 << endl; //12.3457`
- Để định dạng dữ liệu xuất, ta phải thêm chỉ thị:
#include <iomanip>
 - Ví dụ hiển thị 2 chữ số lẻ:
`cout << fixed << setprecision(2)
<< 12.345678 << endl; //12.35`
 - Quy định chiều rộng cột hiển thị:
`cout << "0123456789" << endl; //0123456789
cout << setw(10) << 12.345678 << endl; //12.3457`

Bài tập

1. Viết chương trình nhập vào độ Fahrenheit, chuyển sang độ Celsius và in ra kết quả.
2. Viết chương trình nhập hai cạnh của một hình chữ nhật, in ra màn hình diện tích và chu vi của hình chữ nhật đó
3. Viết chương trình nhập vào hai cạnh đáy và chiều cao của hình thang, in ra màn hình diện tích hình thang
4. Viết chương trình nhập vào giá trị một đơn vị đo tính theo foot (ft), sau đó in ra giá trị tương ứng tính theo:
 - yard (yd)
 - inch (in)
 - centimeter (cm)
 - meter (m)
 - Biết rằng $1 \text{ ft} = 12 \text{ in.}$, $1 \text{ yd} = 3 \text{ ft}$, $1 \text{ in.} = 2.54 \text{ cm}$, $1 \text{ m} = 100 \text{ cm}$.
5. Viết chương trình nhập vào giá trị một đơn vị trọng lượng tính theo ounce (oz), sau đó in ra giá trị tương ứng tính theo:
 - pound (lb)
 - ton (tấn)
 - gram (g)
 - kilogram (kg)
 - Biết rằng $1 \text{ lb} = 16 \text{ oz}$, $1 \text{ ton} = 2000 \text{ lb}$, $1 \text{ oz} = 28.349523 \text{ g}$, $1 \text{ kg} = 1000 \text{ g}$

37

Q & A

38