

Chương 2: Cấu trúc điều khiển

1

Nội dung chính

1. Biểu thức logic
2. Cấu trúc rẽ nhánh
 - if-else
 - switch
3. Cấu trúc lặp:
 - while, do-while, for

2

Biểu thức logic

MATH SYMBOL	ENGLISH	C++ NOTATION	C++ SAMPLE	MATH EQUIVALENT
<code>=</code>	Equal to	<code>==</code>	<code>x + 7 == 2*y</code>	$x + 7 = 2y$
<code>≠</code>	Not equal to	<code>!=</code>	<code>ans != 'n'</code>	$ans \neq 'n'$
<code><</code>	Less than	<code><</code>	<code>count < m + 3</code>	$count < m + 3$
<code>≤</code>	Less than or equal to	<code><=</code>	<code>time <= limit</code>	$time \leq limit$
<code>></code>	Greater than	<code>></code>	<code>time > limit</code>	$time > limit$
<code>≥</code>	Greater than or equal to	<code>>=</code>	<code>age >= 21</code>	$age \geq 21$

3

Biểu thức logic

- `&&`: và
- Kết hợp nhiều biểu thức logic
- Cú pháp
(biểu_thức_logic_1) && (biểu_thức_logic_2)
- Ví dụ:

```
if ( (score > 0) && (score < 10) )
    cout << "score is between 0 and 10.\n";
else
    cout << "score is not between 0 and 10.\n";
```

4

Biểu thức logic

- `||`: hoặc
- Kết hợp nhiều biểu thức logic
- Cú pháp
(biểu_thức_logic_1) || (biểu_thức_logic_2)
- Ví dụ:
if ((score > 0) || (score < 10))
 cout << " score is not between 0 and 10.\n ";
else
 cout << " score is between 0 and 10.\n ";

5

Biểu thức logic

- Kiểu dữ liệu bool
 - true hoặc false
- bool chấp nhận giá trị kiểu int
 - 0: false
 - Ngược lại: true

6

Biểu thức logic

AND

Exp_1	Exp_2	Exp_1 & Exp_2
true	true	true
true	false	false
false	true	false
false	false	false

NOT

Exp	!(Exp)
true	false
false	true

OR

Exp_1	Exp_2	Exp_1 Exp_2
true	true	true
true	false	true
false	true	true
false	false	false

7

::	Scope resolution operator
.	Dot operator
->	Member selection
[]	Array indexing
()	Function call
++	Postfix increment operator (placed after the variable)
--	Postfix decrement operator (placed after the variable)
++	Prefix increment operator (placed before the variable)
--	Prefix decrement operator (placed before the variable)
!	Not
-	Unary minus
+	Unary plus
*	Dereference
&	Address of
new	Create (allocate memory)
delete	Destroy (deallocate)
delete[]	Destroy array (deallocate)
sizeof	Size of object
()	Type cast
*	Multiply
/	Divide
%	Remainder (modulo)
+	Addition
-	Subtraction
<<	Insertion operator (console output)
>>	Extraction operator (console input)

Highest precedence
(done first)

Lower precedence
(done later)

8

<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal
!=	Not equal
&&	And
	Or
=	Assignment
+=	Add and assign
-=	Subtract and assign
*=	Multiply and assign
/=	Divide and assign
%=	Modulo and assign
? :	Conditional operator
throw	Throw an exception
,	Comma operator

Lowest precedence (done last)

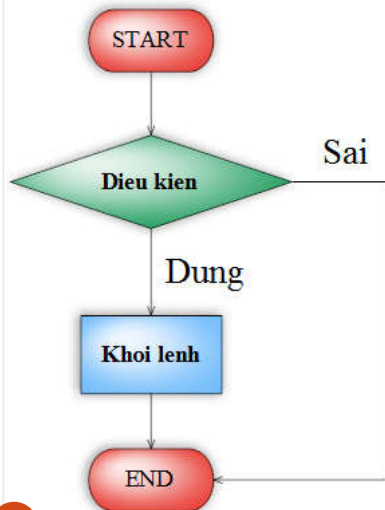
- Xác định giá trị của các biểu thức logic sau, biết count=0, limit=10;

```

a. (count == 0) && (limit < 20)
b. count == 0 && limit < 20
c. (limit > 20) || (count < 5)
d. !(count == 12)
e. (count == 1) && (x < y)
f. (count < 10) || (x < y)
g. !( ((count < 10) || (x < y)) && (count >= 0) )
h. ((limit/count) > 7) || (limit < 20)
i. (limit < 20) || ((limit/count) > 7)
j. ((limit/count) > 7) && (limit < 0)
k. (limit < 0) && ((limit/count) > 7)
l. (5 && 7) + (!6)

```

Cấu trúc rẽ nhánh Câu lệnh if đơn giản



- Cú pháp:

```

if (<điều kiện>)
{
    <Khởi lệnh>
}

```
- Ví dụ:

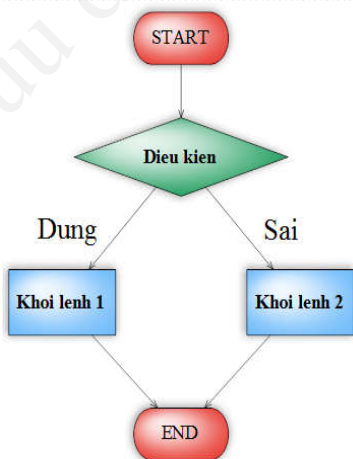
```

int num = 4;
if(num%2==0)
{
    cout<<num<<" là số chẵn"<<endl;
}

```

11

Cấu trúc rẽ nhánh Câu lệnh if else



- Cú pháp:

```

if(<điều kiện>)
{
    <khởi lệnh 1>
} else
{
    <khởi lệnh 2>
}

```
- Ví dụ:

```

int num = 4;
if(num%2==0)
{
    cout<<num<<" là số chẵn"<<endl;
} else
{
    cout<<num<<" là số lẻ"<<endl;
}

```

12

Cấu trúc rẽ nhánh Câu lệnh switch case

- Cú pháp

```
switch (<biểu thức nguyên>)
{
    case nhan_1: <k.lenh1>; break;
    case nhan_2: <k.lenh2>; break;
    case nhan_n: <k.lenh>; break;
    default: <k.lenh>;
}
```

13

Cấu trúc rẽ nhánh Câu lệnh switch case

<pre>#include <iostream> using namespace std; int main() { int x = 2; switch (x) { case 1: cout << "x is 1"; break; case 2: cout << "x is 2"; break; default: cout << "value of x unknown"; } }</pre>	<pre>#include <iostream> using namespace std; int main() { if (x == 1) { cout << "x is 1"; } else if (x == 2) { cout << "x is 2"; } else { cout << "value of x unknown"; } }</pre>
---	--

14

Cấu trúc rẽ nhánh Câu lệnh switch case

```
#include <iostream>
using namespace std;
int main()
{
    float num1, num2; char op;
    cout<<"Go vao so, toan tu, so"<<endl;
    cin>>num1>>op>>num2;
    switch (op)
    {
        case '+': cout<< num1+num2;    break;
        case '-': cout<< num1-num2;    break;
        case '*': cout<< num1*num2;    break;
        case '/': cout<< num1/num2;    break;
        default : cout<< "Toan tu la, khong biet";
    }
}
```

15

Toán tử điều kiện

- Cú pháp:
biểu_thức_logic? giá_trị_1: giá_trị_2;
- Ví dụ:
int n1 = 3, n2=5;
int max = **(n1 > n2) ? n1 : n2;**

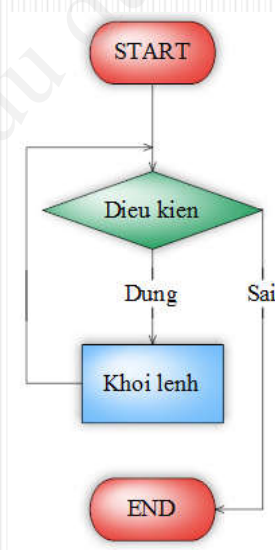
16

Cấu trúc lặp

- while
- do ... while
- for

17

Cấu trúc lặp while



18

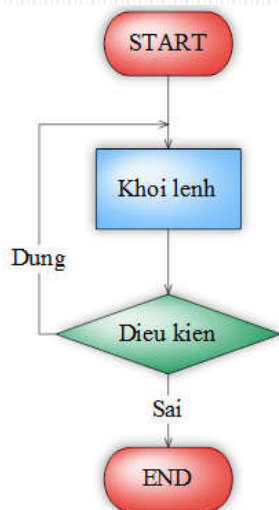
- Cú pháp:
while (điều_kiện)
 {
 thân_chương_trình
 }
- Ví dụ: In bảng mã ASCII

```

int main ()
{
    int n=0;
    while ( n <= 255)
    {
        cout<<char(n)<<" có mã ASCII là "<<n;
        n ++;
    }
}

```

Cấu trúc lặp do ... while



19

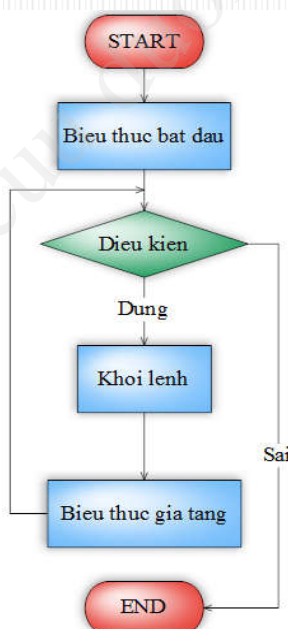
- Cú pháp;

```
do
{
    <khởi lệnh>
} while (<điều kiện>);
```

- Ví dụ:

```
int main ()
{
    int n=0;
    do
    {
        cout<<char(n)<<" co ma ASCII la "<<n;
        n ++;
    } while (n <= 255);
}
```

Cấu trúc lặp for



20

- Cú pháp:

```
for (<biểu thức bắt đầu>; <điều kiện>;
    <biểu thức gia tăng>){
    <khởi lệnh>
}
```

- Ví dụ: tính tổng 10 số tự nhiên đầu tiên

```
int main ()
{
    int i, sum=0;
    for (i=0; i<= 9; i++)
    {
        sum+=i;
    }
}
```

Toán tử ,

- Ví dụ:
first = (first = 2, second = first + 1);
 - second = 3
 - first = 3
- Thực hiện danh sách các biểu thức, trả về giá trị của biểu thức cuối cùng
- Thường dùng cho vòng lặp for

21

break, continue

- break; kết thúc vòng lặp ngay tức khắc
- continue; bỏ qua phần còn lại của lệnh lặp
- Hạn chế sử dụng

22

break, continue

```
#include <iostream>
using namespace std;
int main()
{
    for (int n=10; n>0; n--)
    {
        cout << n << ", ";
        if (n==3)
        {
            cout << "Stop!";
            break;
        }
    }
}
```

```
#include <iostream>
using namespace std;
int main()
{
    for (int n=10; n>0; n--)
    {
        if (n==5)
            continue;
        cout << n << ", ";
    }
    cout << "liftoff!\n";
}
```

23