



C++ Course

Buổi 1: Kiểu dữ liệu, cấu trúc điều khiển, toán tử





NỘI DUNG CHÍNH

01

BIẾN - VARIABLE

Biến là gì?

02

Kiểu dữ liệu - DATATYPE

Có bao nhiêu kiểu dữ liệu

03

CẤU TRÚC Rẽ NHÁNH

If...else...
Switch...case

04

TOÁN TỬ - OPERATOR

Cộng trừ nhân chia
Tăng giảm,....



Biến – Variable

Khái niệm:

- Biến = Vùng nhớ + Định danh.
- Biến dùng để lưu trữ giá trị.
- Thông qua định danh chúng ta có thể truy cập vào biến và sử dụng giá trị nó đang nắm giữ.
- Ví dụ:

```
int a; // Khai báo một biến, có định danh là a, dùng để lưu trữ một giá trị nguyên, độ dài 4 byte  
float b = 2.135; // Khởi tạo một biến, có định danh là b lưu trữ số thực, có sẵn giá trị là 2.135, độ dài 4 byte
```
- Vậy thì int, float, double, char, bool, ... là cái gì ???
- Kiểu dữ liệu nguyên thủy:
 - INT : tạo ra một biến lưu trữ số nguyên
 - FLOAT, DOUBLE : tạo ra một biến lưu trữ số thực dấu chấm động.
 - CHAR : tạo ra một biến lưu trữ một ký tự.
 - BOOL : tạo ra một biến lưu trữ giá trị TRUE hoặc FALSE.





“Kiểu dữ liệu là thứ tồn tại duy nhất,
những thứ khác có hay không có,
không quan trọng”

—Phan Hải said



DATATYPE

Kiểu dữ liệu là một cách phân loại dữ liệu cho trình biên dịch hoặc thông dịch hiểu các lập trình viên muốn sử dụng dữ liệu.



A PICTURE IS WORTH A THOUSAND WORD





Phân loại kiểu dữ liệu

Cơ bản - Primitive

Đã được trình biên dịch
quy định, định nghĩa sẵn

Int – số nguyên
Float – số thực
Double – số thực
Char – ký tự
Bool - boolean

Phức tạp - Complexity

Do lập trình viên quy
định, định nghĩa

Struct – cấu trúc
Union – hợp
Enum – liệt kê
Class – lớp



01

Primitive datatype





Primitive datatype

Kiểu dữ liệu nguyên thủy

Kiểu dữ liệu nguyên thủy được quy định sẵn:

- Kích cỡ cố định
- Lưu trữ loại giá trị riêng
- Không bị thay đổi



Ví dụ: `int a = 10;`

- Định danh `a` + kiểu dữ liệu `int` => biến nguyên `a`
- Kích cỡ là 4 byte
- Có giá trị khởi tạo là 10
- Kiểu dữ liệu nguyên thủy:
 - `INT` : tạo ra một biến lưu trữ số nguyên
 - `FLOAT`, `DOUBLE` : tạo ra một biến lưu trữ số thực dấu chấm động.
 - `CHAR` : tạo ra một biến lưu trữ một ký tự.
 - `BOOL` : tạo ra một biến lưu trữ giá trị `TRUE` hoặc `FALSE`.

02

Cấu trúc điều khiển



Cấu trúc điều khiển



If...else

Cấu trúc rẽ nhánh

Cú pháp:
If(conditional)
{code}
Else
{code}
....



Switch...case

Cấu trúc lựa chọn:

Switch(expression)
Case const 1:
Case const 2:
Case const 3:
Case const 4:
...

If(conditional) {} else {}

1

Kiểm tra điều kiện

Trước khi xác định tiếp theo cần làm gì, phải kiểm tra điều kiện

2

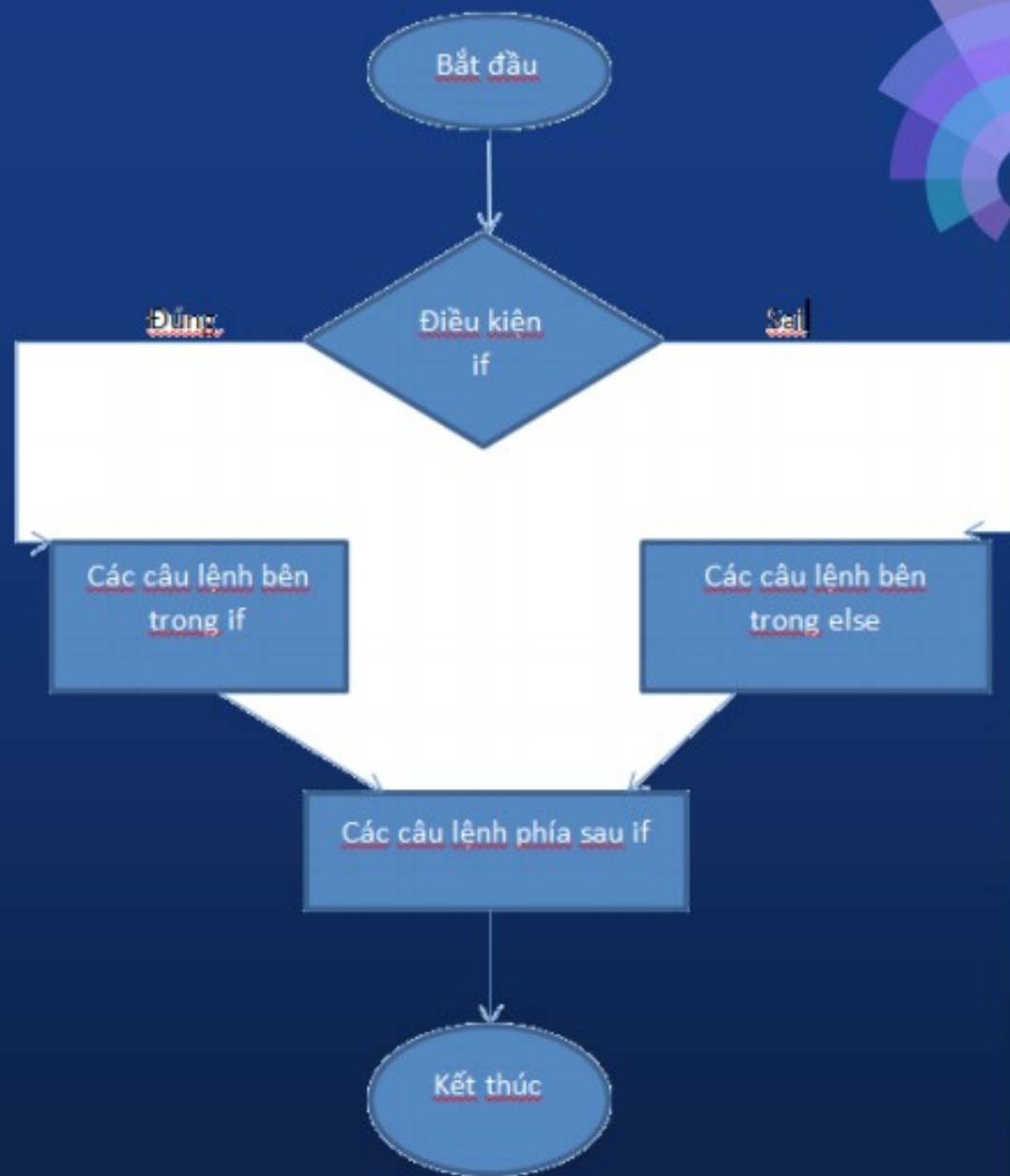
Điều kiện đúng

Điều kiện đúng, thực hiện nội dung trong if

3

Điều kiện sai

Điều kiện sai, thực hiện nội dung của else





Switch ... case

Tiến hành:

- Kiểm tra biểu thức
- Đảm bảo các case là constant
- Kiểm tra trùng giá trị.
- Thực hiện từ trên xuống dưới khi có một giá trị trùng.
- Default được thực thi khi không có trùng giá trị.

```
switch(<bieuthuc>|
{
    case <gia tri 1>:
    {
        //xử lý khi biểu thức bằng giá trị 1
        break;
    }
    case <gia tri 2>:
    {
        //xử lý khi biểu thức bằng giá trị 2
        break;
    }
    case <gia tri 3>:
    {
        //xử lý khi biểu thức bằng giá trị 3
        break;
    }
    ~~~
    default:
        //xử lý khi biểu thức không bằng các giá trị trên
}
```



Bài tập luyện tập

Luyện tập mỗi ngày
nâng cao trình độ

- Nhập số nguyên vào, kiểm tra chẵn lẻ
- Nhập ký tự vào, kiểm tra hoa - thường
- Nhập số nguyên vào, kiểm tra số âm - dương

03

Operator – toán tử





Toán tử là gì

Khái niệm

Toán tử - operator là hàm, chức năng dùng để thao tác các giá trị với nhau



Đặc điểm

Trong C++, là các ký tự được quy định sẵn: + - * / % -- ++ &&



Cách dùng

Phép toán = Toán hạng + Toán tử



Phân loại

Một ngôi : ++ -- ! & - ::

Hai ngôi : + - * /

Ba ngôi : ?:



Toán tử một ngôi

Tăng	Giảm	Âm
++	--	-
Tăng giá trị thêm 1	Giảm giá trị đi 1	Chuyển thành số âm





Toán tử 2 ngôi

Số học

— ● + - * / %

Quan hệ

— ● < > <= >= == !=

Logic

— ● && ||

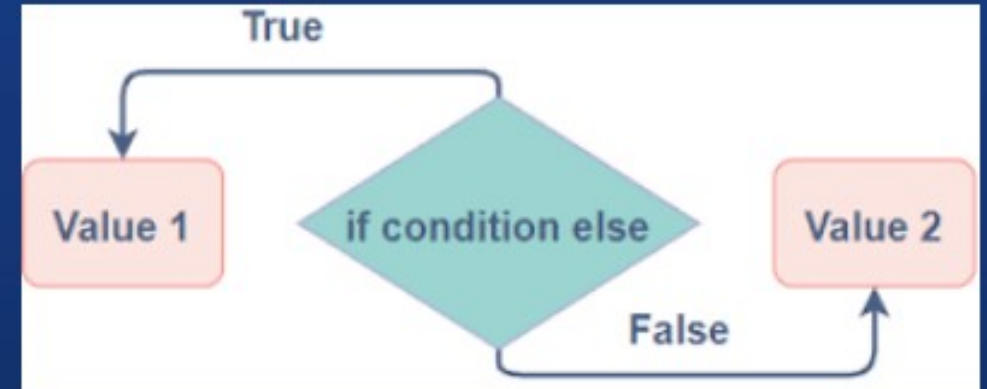
Gán

— ● += -= *= /= %=



Toán tử 3 ngôi

- Cú pháp: conditional ? Statement 1: Statement 2;
- Ví dụ:
 - Kiểm tra điều kiện
 - Đúng thì thực hiện statement 1.
 - Sai thì thực hiện statement 2.



```
C++ work.cpp > main()
1  #include <stdio.h>
2  int main()
3  {
4      int a = 5;
5      printf("%d", a < 10 ? 1 : 0);
6      return 0;
7  }
```

```
E:\C_CPP>g++ work.cpp -o work.exe && work.exe
1
E:\C_CPP>
```

04

Luyện tập



Tổng kết

