



3. Kiểu dữ liệu

🕒 Thời gian tạo	@August 13, 2023 6:48 PM
📄 Loại bài giảng	Lý thuyết

1. Đặt vấn đề

- Hàng ngày, chúng ta gặp các vấn đề khác nhau. Với mỗi một vấn đề, chúng ta sẽ có cách xử lý khác nhau.
- Tương tự như vậy, máy tính chúng ta sử dụng cũng phải đối mặt với nhiều loại thông tin khác nhau. Mỗi một loại thông tin, máy tính sẽ đưa ra các phương pháp xử lý để cho ra kết quả như mong muốn.
- Như vậy, kiểu dữ liệu ra đời, nhằm xác định rõ loại dữ liệu để từ đó có thể xác định rõ ràng các phương án xử lý dữ liệu tương ứng.

2. Phân loại các kiểu dữ liệu

- Kiểu dữ liệu nguyên thủy (Primitive data types): là những kiểu dữ liệu được chính ngôn ngữ lập trình định nghĩa/quy định sẵn.

- Kiểu dữ liệu tự chọn/tự thiết lập (custom/user-defined data types): là những kiểu dữ liệu do chính chúng ta tạo ra, dựa trên các kiểu dữ liệu có sẵn.

⚠ Bài viết này sẽ tập trung vào kiểu dữ liệu nguyên thủy (sau đây sẽ gọi ngắn gọn là **primitives**). Kiểu dữ liệu user-defined sẽ được đề cập trong những bài viết sau.

3. Các kiểu dữ liệu nguyên thủy (primitives) trong C++

3.1. Các kiểu dữ liệu số nguyên (integer):

Kiểu dữ liệu	Kích thước (byte)	Khoảng giá trị
<code>char</code>	1	<code>-128</code> → <code>127</code> (hoặc $[-2^7, 2^7 - 1]$)
<code>short int</code> (hoặc <code>short</code>)	2	<code>-32768</code> → <code>32767</code> (hoặc $[-2^{15}, 2^{15} - 1]$)
<code>unsigned short int</code> (hoặc <code>unsigned short</code>)	2	<code>0</code> → <code>65535</code> (hoặc $[0, 2^{16} - 1]$)
<code>int</code>	4	$[-2^{31}, 2^{31} - 1]$
<code>unsigned int</code>	4	$[0, 2^{32} - 1]$
<code>long int</code> (hoặc <code>long</code>)	4	(giống <code>int</code>)
<code>unsigned long int</code> (hoặc <code>unsigned long</code>)	4	(giống <code>unsigned int</code>)
<code>long long int</code> (hoặc <code>long long</code>)	8	$[-2^{63}, 2^{63} - 1]$
<code>unsigned long long int</code> (hoặc <code>unsigned long long</code>)	8	$[0, 2^{64} - 1]$

- Từ bảng trên ta có khoảng giá trị chung cho các kiểu dữ liệu số nguyên theo công thức (x là kích thước của kiểu dữ liệu **tính bằng bit**):

$$\begin{cases} \text{signed:} & [-2^{x-1}, 2^{x-1} - 1] \\ \text{unsigned:} & [0, 2^x - 1] \end{cases}$$

- Như vậy, mỗi 1 kiểu dữ liệu số nguyên sẽ có tổng cộng 2^x giá trị.

3.2. Các kiểu dữ liệu số thực:

Kiểu dữ liệu	Kích thước (byte)	Khoảng giá trị
<code>float</code>	4	$[-3.4 \times 10^{38}, 3.4 \times 10^{38}]$
<code>double</code>	8	$[-1.7 \times 10^{308}, 1.7 \times 10^{308}]$
<code>long double</code>	12	$[-1.1 \times 10^{4932}, 1.1 \times 10^{4932}]$

- Kiểu dữ liệu logic:

Kiểu dữ liệu	Kích thước (byte)	Khoảng giá trị
<code>bool</code>	1	Chỉ nhận 2 giá trị: <code>true</code> và <code>false</code> (hoặc <code>1</code> và <code>0</code>)

4. Sử dụng với biến:

 Xem ví dụ code ở bài số 2, phần 3 về biến

- Một số lưu ý:
 - Một khi đã khai báo 1 biến với một kiểu dữ liệu, chỉ được gán/gán lại với giá trị thuộc kiểu dữ liệu đó (hoặc “thuộc phạm vi cho phép”)
 - Không nên gán giá trị nằm ngoài khoảng giá trị của kiểu dữ liệu.

5. Xác định kích thước của kiểu dữ liệu với lệnh `sizeof`

- Cách sử dụng:

```
sizeof(kiểu_dữ_liệu)
```

- Paste đoạn code sau:

```
#include<iostream>
using namespace std;

int main() {
    cout << "Size of char is: " << sizeof(char) << "bytes" << endl;
    cout << "Size of int is: " << sizeof(int) << "bytes" << endl;
    cout << "Size of long long is: " << sizeof(long long) << "bytes" << endl;
    cout << "Size of float is: " << sizeof(float) << "bytes" << endl;
    cout << "Size of bool is: " << sizeof(bool) << "bytes" << endl;
}
```

- Kết quả của đoạn code khi chạy là:

```
Size of char is: 1 bytes
Size of int is: 4 bytes
Size of long long is: 8 bytes
Size of float is: 4 bytes
Size of bool is: 1 bytes
```

⚠ Kết quả trên có thể khác tùy thuộc vào máy tính

6. Thư viện kiểm tra giới hạn giá trị của kiểu dữ liệu: `<climits>`

- Thư viện này cung cấp những biến hằng (constants) biểu thị giá trị lớn nhất hoặc nhỏ nhất của các kiểu dữ liệu số nguyên.
- Cách sử dụng:

```
#include<iostream>
#include<climits>
using namespace std;

int main() {
    cout << "Maximum of int is: " << INT_MAX << endl;
    cout << "Minimum of long is: " << LONG_MIN << endl;
}
```

- Tương tự, ta có thể thử với các constant khác nằm trong thư viện này. Click vào đường link này để xem thêm: <https://cplusplus.com/reference/climits/?kw=climits>
- Với kiểu dữ liệu số thực (`float` , `double`), xem thêm thư viện `<cmath>` .