Bài 8: KIỂU KÝ TỰ TRONG C++ (CHARACTER)

Xem bài học trên website để ủng hộ Kteam: Kiểu ký tự trong C++ (Character)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Ở bài học trước, bạn đã nắm được 2 kiểu dữ liệu cực kỳ quan trọng: SỐ NGUYÊN & SỐ CHẤM ĐỘNG TRONG C++ (Integer, Floating point), và đã biết được những kinh nghiệm cũng như những lỗi thường gặp khi sử dụng nó.

Hôm nay, mình sẽ giới thiệu cho các bạn một kiểu dữ liệu khá quan trọng trong lập trình: **Kiểu ký tự trong C++ (Character)**

Nội dung:

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- BIẾN TRONG C++ (Variables)
- SỐ NGUYÊN & SỐ CHẤM ĐỘNG TRONG C++ (Integer, Floating point)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Tổng quan về kiểu ký tự (Character)
- Khai báo, khởi tạo và gán giá trị một biến ký tự
- In ký tự ra màn hình
- In ký tự từ số nguyên và ngược lại (Casting)
- Escape sequences
- Newline '\n' và std::endl
- Dấu nháy đơn 'K' và dấu nháy kép "Kteam" trong C++

Tổng quan về kiểu ký tự (Character)

Ở bài học trước, bạn đã học về kiểu dữ liệu số học Integer và Floating point, những kiểu này dùng để giải quyết các bài toán trong chương trình, và lưu trữ nó. Một chương trình nếu chỉ hiển thị số thì người dùng có nhìn vào cũng không hiểu được, lúc này bạn sẽ cần những biến để lưu trữ ký tự, hay những câu thông báo cho người dùng. Để làm được điều đó, bạn cần học qua kiểu char trong C++.

Type	Size	Range (Min)	Range (Max)
signed (char)	1 byte	-128	127
unsigned (char)	1 byte	0	255

Trong C++, char là một kiểu dữ liệu đặc biệt, nó vừa là kiểu số nguyên, cũng vừa là kiểu ký tự. Do đó, kiểu char tuân thủ tất cả các quy tắc của một số nguyên bình thường (+-*/...).

> Một biến kiểu char sẽ có độ lớn **1 bytes (8 bits)**, **dùng để lưu trữ một ký tự trong bảng mã ASCII** như (a, b, c, ... 1, 2, 3, ...)

ASCII (**American Standard Code for Information Interchange** - Chuẩn mã trao đổi thông tin Hoa Kỳ) là bộ kí tự và bộ mã kí tự dựa trên bảng chữ cái La Tinh trong tiếng Anh. Bảng bên dưới mô tả đầy đủ ký tự ASCII:

Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol
0	NUL (null)	32	(space)	64	@	96	`
1	SOH (start of header)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(	72	H	104	h
9	HT (horizontal tab)	41	)	73	I	105	i
10	LF (line feed/new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (form feed / new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (data control 1)	49	1	81	Q	113	q
18	DC2 (data control 2)	50	2	82	R	114	r
19	DC3 (data control 3)	51	3	83	S	115	s
20	DC4 (data control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of transmission block	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y

26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[	123	{
28	FS (file separator)	60	<	92	\	124	|
29	GS (group separator)	61	=	93	]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL (delete)

Có 2 loại ký tự trong bảng mã ASCII:

- Ký tự không in được (ký tự điều kiển) có mã từ 0 -> 31 và 127 bao gồm các ký tự dùng để điều khiển hay dùng như: backspace (7), new line (10), escape (27), bell (7) ...Có 2 loại ký tự trong bảng mã ASCII:
- Ký tự in được có mã từ 32 -> 126 bao gồm các chữ cái (a, b, c, ...), số (1, 2, 3, ...), toán tử (+, -, ...), dấu câu (~, {, ^ ...)

Bảng mã ASCII chuẩn có 128 ký tự, bảng mã ASCII mở rộng có 255 ký tự.

Khai báo, khởi tạo và gán giá trị một biến ký tự

Để khai báo, khởi tạo và gán giá trị cho một biến kiểu ký tự trong C++, bạn sử dụng từ khóa **char:**

```
char ch1{ 'K' };// khởi tạo biến character với ký tự 'K' (mã ASCII 75)
char ch2{ 75 };     // khởi tạo biến character với mã ASCII 75 (ký tự 'K')
char ch3(75);       // khởi tạo biến character với mã ASCII 75 (ký tự 'K')
char ch4 = 'K';     // khởi tạo biến character với ký tự 'K' (mã ASCII 75)
char ch5;           // khai báo biến kiểu character
ch1 = 75;           // gán mã ASCII 75 (ký tự 'K') cho biến character
```

- **Chú ý 1:** Vì bản chất của kiểu **char** cũng là số nguyên, nên khi khởi tạo hoặc gán giá trị cho biến kiểu **char,** bạn hoàn toàn có thể dùng số nguyên (mã ASCII) hoặc ký tự.

```
char ch1{ 75 };     // mã ASCII 75 (ký tự 'K')
char ch2{ 'K' };    // ký tự 'K' (mã ASCII 75)
```

- **Chú ý 2:** Ký tự số nguyên không giống với mã ASCII số nguyên, bạn cần phân biệt điều này để tránh nhầm lẫn khi khởi tạo một biến character.

Hai cách khởi tạo bên dưới hoàn toàn khác nhau:

```
char ch1{ '7' };    // khởi tạo biến character với ký tự '7' (mã ASCII 55)
char ch2{ 7 };      // khởi tạo biến character với mã ASCII 7 (tiếng "beep")
```

In ký tự ra màn hình

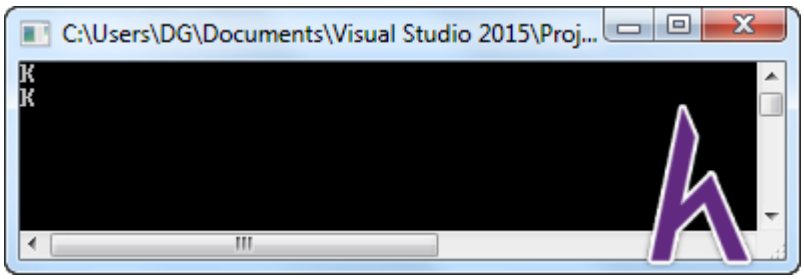
Dù khi khởi tạo hoặc gán giá trị cho biến kiểu **char,** bạn hoàn toàn có thể dùng số nguyên (mã ASCII) hoặc ký tự. Nhưng khi in ra màn hình, nó hiển thị một ký tự ASCII thay vì một số nguyên.

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

int main()
{
	char ch1{ 75 };     // mã ASCII 75 (ký tự 'K')
cout << ch1 << endl;      // ký tự 'K' với mã ASCII 75 được hiển thị ra màn hình

char ch2{ 'K' };    // ký tự 'K' (mã ASCII 75)
cout << ch2 << endl;      // ký tự 'K' trực tiếp được hiển thị ra màn hình
	return 0;
}
```

Ở chương trình trên, 2 biến **ch1**, **ch2** đều cho kết quả **'K'** như nhau:



In ký tự từ số nguyên và ngược lại (Casting)

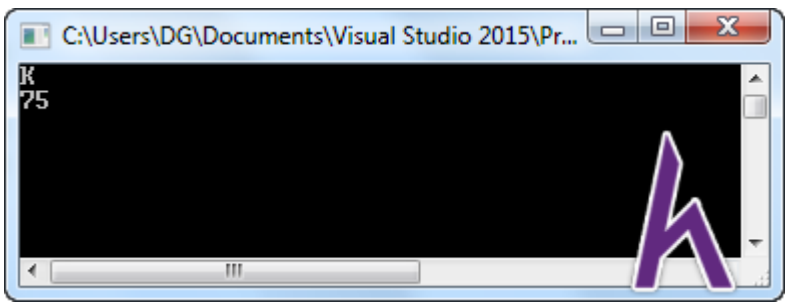
Vì **char là một kiểu dữ liệu đặc biệt, nó vừa là kiểu số nguyên, cũng vừa là kiểu ký tự**. Đồng nghĩa với việc bạn có thể in ra màn hình ký tự hoặc mã ASCII của nó:

```
#include <iostream>
#include <iomanip>		// for std::setprecision()
using namespace std;

int main()
{
	int n{ 75 };
	cout << static_cast<char>(n) << endl; // in ký tự với mã ASCII 75

	char ch{ 'K' };
	cout << static_cast<int>(ch) << endl; // in mã ASCII của ký tự 'K'
	return 0;
}
```

Outputs:



Chương trình trên sử dụng kỹ thuật ép kiểu **static cast** trong C++ để in một ký tự từ một số nguyên và ngược lại. Cú pháp ép kiểu **static cast**:

```
static_cast<Type-id>(expression)
```

Lệnh **static cast** sẽ chuyển đổi giá trị của **expression** thành một giá trị có kiểu **Type-id** (char, int, double, ...), và trả về giá trị đó.

- **Chú ý 1:** Bản thân **expression** sẽ không bị ảnh hưởng sau khi ép kiểu (n vẫn là 75, ch vẫn là 'K').
- **Chú ý 2:** Khi bạn ép kiểu từ int sang char, nếu giá trị int của bạn lớn hơn giới hạn kiểu char, vấn đề tràn số sẽ xảy ra.

Phạm vi bài học này chỉ đề cập cơ bản về ép kiểu và static_cast, vấn đề này sẽ được giới thiệu chi tiết hơn ở bài: ÉP KIỂU TƯỜNG MINH TRONG C++ (Explicit type conversion)

Escape sequences

Trong C++, **có một số ký tự mang ý nghĩa đặc biệt** dùng trong chuỗi hay mảng ký tự **gọi là Escape sequence.** Một **escape sequence** có cấu trúc gồm một dấu **'\' (backslash),** sau đó là một ký tự hoặc một số.

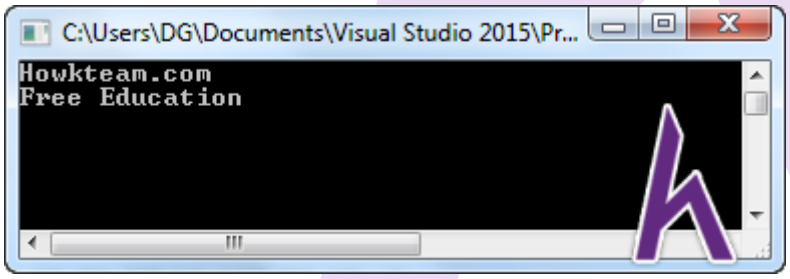
Bảng bên dưới liệt kê các Escape sequence trong C++:

Name	Symbol	Meaning
Alert	\a	Makes an alert, such as a beep
Backspace	\b	Moves the cursor back one space
Formfeed	\f	Moves the cursor to next logical page
Newline	\n	Moves cursor to next line
Carriage return	\r	Moves cursor to beginning of line
Horizontal tab	\t	Prints a horizontal tab
Vertical tab	\v	Prints a vertical tab
Single quote	\'	Prints a single quote
Double quote	\"	Prints a double quote
Backslash	\\	Prints a backslash
Question mark	\?	Prints a question mark
Octal number	\(number)	Translates into char represented by octal
Hex number	\x(number)	Translates into char represented by hex number

Một số ví dụ về **escape sequence** thường gặp:

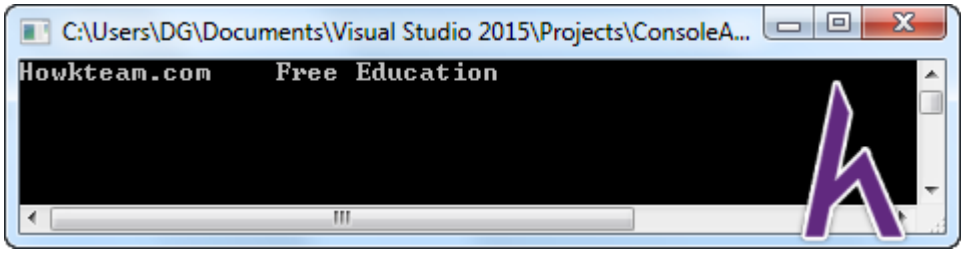
```
std::cout << "Howkteam.com\nFree Education"; // '\n' di chuyển con trỏ xuống một dòng mới
```

Outputs:




```
std::cout << "Tab 1\tTab 2 "; // '\t' di chuyển con trỏ 1 tab
```

Outputs:



Vẫn còn rất nhiều escape sequence, bạn hãy tự mình khám phá nhé.

Newline '\n' và std::endl

Ở những bài học trước, bạn sử dụng **std::endl** để xuống dòng khi in một chuỗi, nhưng sau khi học về **Escape sequence**, bạn lại biết thêm một cách để xuống dòng nữa là sử dụng escape sequence **'\n'.**

Nếu bạn viết một chương trình như bên dưới và sử dụng cả 2 cách, bạn sẽ có được kết quả như nhau:

```
std::cout << "HowKteam.com" << std::endl;
std::cout << "Free education\n";
```

Tuy nhiên, 2 cách này có thực sự giống nhau? **Câu trả lời là không**, **bản chất của std::endl** được thể hiện ở 2 câu lệnh bên dưới:

```
std::cout << "HowKteam.com" << std::endl;

// Tương đương với:

std::cout << "HowKteam.com\n" << std::flush;
```

Trong C++, **output stream thường dùng buffer**, nghĩa là **output data** sẽ **được lưu vào một vùng nhớ đệm**, và **output data sẽ được gửi đến output device** vào thời điểm thích hợp (vì lý do hiệu suất). Với **std::endl** sẽ **xóa output buffer** mỗi khi nó được gọi, trong khi **'\n' thì không**.

Vậy, khi nào nên sử dụng **std::endl** và **'\n':**

- **Nên sử dụng std::endl** khi bạn cần **đảm bảo output** của bạn có **ngay lập tức** (Vd: khi viết một record vào một file, hoặc khi update một thanh tiến trình). Nhưng nên hạn chế sử dụng **std::endl** khi làm việc với file I/O để tránh việc phải flush buffer liên tục dẫn đến việc phải truy cập các file I/O thường xuyên (giảm hiệu suất).
- Ngoài ra, những trường hợp khác nên sử dụng **'\n'**.

Dấu nháy đơn 'K' và dấu nháy kép "Kteam"

Trong suốt bài học này, bạn có thể thấy các **character luôn được đặt trong dấu nháy đơn** (Vd: 'K', 't', 'e', 'a', 'm'). Một biến kiểu char sẽ lưu một ký tự, nếu bạn khởi tạo nhiều hơn 1 ký tự thì sẽ nhận được một thông báo lỗi như câu lệnh bên dưới:

```
char ch{ 'Kteam' };      // Câu lệnh này sẽ cho ra một thông báo lỗi
```

Những ký tự đặt trong dấu nháy kép gọi là một chuỗi ký tự (Vd: "HowKteam.com"), **chuỗi ký tự là một tập hợp của các ký tự** được đặt trong dấu nháy kép. Câu lệnh bên dưới thể hiện "HowKteam.com" là một chuỗi ký tự (string).

```
std::cout << "HowKteam.com";      // HowKteam.com là một chuỗi ký tự
```

Phạm vi bài học này chỉ đề cập đến ký tự, chuỗi ký tự (string) là tương đối phức tạp. Mình sẽ giới thiệu về chuỗi ký tự (string) trong bài CHUỖI KÝ TỰ TRONG C++ (String).

Kết luận

Qua bài học này, bạn đã nắm được Kiểu ký tự trong C++ (Character), và đã biết được những kinh nghiệm cũng như thắc mắc liên quan đến kiểu ký tự khi sử dụng nó.

Trong C++, ngoài kiểu integer, floating point, character, thì vẫn còn một kiểu dữ liệu cực kỳ quan trọng. Bạn sẽ được học nó trong bài học tiếp theo: KIỂU LUẬN LÝ TRONG C++ (Boolean)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".