

基本数据类型

1.char 2. int 3. short 4. long 5. float 6.double 7.struct(自定义数据类型) 8. enum(枚举) 9.union(联合) 10.class(类) 11.template (类模板) 12.reference(引用) 13.void类型 14.指针类型

类型 (64位系统下)	位	范围
char	1 个字节	-128 到 127 或者 0 到 255
unsigned char	1 个字节	0 到 255
signed char	1 个字节	-128 到 127

类型 (64位系统下)	位	范围
bool	1 个字节	非整数0即为true(-1、“0”...都是true，只有整数0为false)
int	4 个字节	-2147483648 到 2 147 483 647 (10^9) (2^{31}) (压缩DP)
unsigned int	4 个字节	0 到 4294967295
signed int	4 个字节	-2147483648 到 2147483647
short int	2 个字节	-32768 到 32767
unsigned short int	2 个字节	0 到 65,535
signed short int	2 个字节	-32768 到 32767
long long int	8 个字节	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
signed long long int	8 个字节	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
unsigned long long int	8 个字节	0 到 18,446,744,073,709,551,615 ($2^{64} - 1$)
float	4 个字节	精度型占4个字节 (32位) 内存空间, +/- 3.4e +/- 38 (~7 个数字)
double	8 个字节	双精度型占8 个字节 (64位) 内存空间, +/- 1.7e +/- 308 (~15 个数字)
long double	16 个字节	长双精度型 16 个字节 (128位) 内存空间, 可提供18-19位有效数字。
wchar_t	2 或 4 个字节	1 个宽字符

signed
单词意思: adj. 有符号的

unsigned
单词意思: adj. 无符号的

signed和unsigned可以用来修饰char型和int型 (包括long int) , 不能修饰bool、float、double和long double。

运算符优先级

运算符优先级: ! > 算术运算符 > 关系运算符 > (&& ||) > 条件运算符> 赋值运算符 > 逗号运算符。

详细表:

优先级	操作符	描述	例子	结合性
1	() [] -> . :: ++ --	调节优先级的括号操作符 数组下标访问操作符 通过指向对象的指针访问成员的操作符 通过对象本身访问成员的操作符 作用域操作符 后置自增操作符 后置自减操作符	(a + b) / 4; array[4] = 2; ptr->age = 34; obj.age = 34; Class::age = 2; for(i = 0; i < 10; i++) ... for(i = 10; i > 0; i--) ...	从左到右
2	! ~ ++ -- - + * & (type) sizeof	逻辑取反操作符 按位取反(按位取补) 前置自增操作符 前置自减操作符 一元取负操作符 一元取正操作符 解引用操作符 取地址操作符 类型转换操作符 返回对象占用的字节数操作符	if(!done) ... flags = ~flags; for(i = 0; i < 10; ++i) ... for(i = 10; i > 0; --i) ... int i = -1; int i = +1; data = *ptr; address = &obj; int i = (int) floatNum; int size = sizeof(floatNum);	从右到左
3	->* .*	在指针上通过指向成员的指针访问成员的操作符 在对象上通过指向成员的指针访问成员的操作符	ptr->var = 24; obj.var = 24;	从左到右
4	* / %	乘法操作符 除法操作符 取余数操作符	int i = 2 * 4; float f = 10 / 3; int rem = 4 % 3;	从左到右
5	+ -	加法操作符 减法操作符	int i = 2 + 3; int i = 5 - 1;	从左到右
6	<< >>	按位左移操作符 按位右移操作符	int flags = 33 << 1; int flags = 33 >> 1;	从左到右
7	< <= > >=	小于比较操作符 小于或等于比较操作符 大于比较操作符 大于或等于比较操作符	if(i < 42) ... if(i <= 42) ... if(i > 42) ... if(i >= 42) ...	从左到右
8	== !=	等于比较操作符 不等于比较操作符	if(i == 42) ... if(i != 42) ...	从左到右
9	&	按位与操作符	flags = flags & 42;	从左到右

优先级	操作符	描述	例子	结合性
10	<code>^</code>	按位异或操作符	<code>flags = flags ^ 42;</code>	从左到右
11	<code> </code>	按位或操作符	<code>flags = flags 42;</code>	从左到右
12	<code>&&</code>	逻辑与操作符	<code>if(conditionA && conditionB) ...</code>	从左到右
13	<code> </code>	逻辑或操作符	<code>if(conditionA conditionB) ...</code>	从左到右
14	<code>?:</code>	三元条件操作符	<code>int i = (a > b) ? a : b;</code>	从右到左
15	<code>= += - = *= /= %= &= ^= = <<= >>=</code>	赋值操作符 复合赋值操作符(加法) 复合赋值操作符(减法) 复合赋值操作符(乘法) 复合赋值操作符(除法) 复合赋值操作符(取余) 复合赋值操作符(按位与) 复合赋值操作符(按位异或) 复合赋值操作符(按位或) 复合赋值操作符(按位左移) 复合赋值操作符(按位右移)	<code>int a = b; a += 3; b -= 4; a *= 5; a /= 2; a %= 3; flags &= new_flags; flags ^= new_flags; flags = new_flags; flags <<= 2; flags >>= 2;</code>	从右到左
16	<code>,</code>	逗号操作符	<code>for(i = 0, j = 0; i < 10; i++, j++) ...</code>	从左到右

从右到左的意思：先计算右边的值再附到左边

例子1:

```

a=b=c+1
先算c + 1
再算b = (c + 1)
再算a = b

```

例子2:

```
int a = 2;
cout << (a == 1 ? 10 : a == 2 ? 20 : 30) << endl;
//最终输出20

a == 1 ? 10 : a == 2 ? 20 : 30看作
a == 1 ? 10 : (a == 2 ? 20 : 30)
计算后面的得
a == 1 ? 10 : 20
```

优先级例子（等学到二维数组再回来看）

```
int a[] = { 1,2,3 };
int b[] = { 4,5,6 };
int c[] = { 7,8,9 };
int* arr[] = { a,b,c };
cout << *arr[1] << endl;
cout << (*arr)[1] << endl;
```

输出： 4
2

合法标识符，合法常量，合法转义字符

一，合法标识符

用户定义的合法标识符需满足以下两个要求：

标识符只能由字母，数字和下划线组成。

标识符不能以数字开头。

不能是c++自带的关键字

二，合法常量

整型常量：

十进制：10

八进制：017（以0开头，不能出现8，9）

十六进制：0xA1（以0x开头）

实型常量：（强调E）

E的前面必须有数字，E的后面必须是整数

字符型：'n','N','\n'

字符串型："abc","123",""

三，合法转义字符

一般转义字符

\a 响铃

\b 退格

\f 换页

等

八进制转义字符
它是由反斜杠\ 和 随后的1~3个八进制数字构成的字符序列
一般形式是'\ddd', d是0-9的数字。

十六进制转义字符
它是由反斜杠\ 和 字母x(或X)及随后的1~2个十六进制数字构成的字符序列
一般形式是'\xhh', h是0-9或A-F内的一个。
八进制转义字符和十六进制转义字符，不在前面加0！

转义字符

转义字符	意义	ASCII码值(十进制)
\a	响铃(BEL)	007
\b	退格(BS)	008
\f	换页(FF)	015
\n	换行(LF)	010
\r	回车(CR)	013
\t	水平制表(HT)	009
\v	垂直制表(VT)	011
\\	反斜杠	092
\?	问号字符	063
\'	单引号字符	039
\"	双引号字符	034
\0	空字符(NULL)	000
\ddd	任意字符	三位八进制
\xhh	任意字符	二位十六进制

“\”加数字（一般是8进制数字）来表示。
常考：没有\c这个转义字符

开发一个C++程序的过程

通常包括编辑、编译、链接、运行和调试等步骤。

