

一. 基本的算术运算符

1. 赋值运算符：找到变量的存储位置，并给他赋值，运算顺序为从右往左。
2. 加法运算符
3. 减法运算符
4. 乘法运算符：注意C有乘法运算符而无指数运算符。
5. 除法运算符：整数除法的小数部分会被趋零截断（C99）；注意计算机不能直接处理不同类型的除法，一般都是将整型转化为浮点型。
6. 求模运算符%：只能用于整数；其正负号根据第一个运算对象决定（第一个是负数，结果也是负数，与第二个运算对象无关）
7. 符号运算符：用来确定字面量或变量的符号，其优先级在各种算术运算符中仅次于括号和自增运算符。
8. 优先级和运算顺序：当运算符共享同一个运算对象时，根据优先级（一个运算对象和多个运算符）；当一个运算符被多个运算对象共享，根据顺序来进行计算（多个运算对象和一个运算符）

二. 特殊的算术运算符

1. sizeof运算符：本质是利用了一个typedef为现有类型创建了一个别名，C头文件系统可以使用typedef把size_t类型作为unsigned int或unsigned long类型。
2. cast operator:将运算符右侧的值转化为圆括号中指定的类型：e.g. (int) 2.7 == 2
3. 自增运算符++和--:作为前缀使用：先递增（减）一再使用；作为后缀使用：先递增（减）一再使用；注意其优先级仅次于括号；
4. **不要自作聪明：如果一个变量出现再函数的多个参数或再一个表达式中多次出现，不要使用自增运算符，这是UB，编译器会自行选择先算哪一个。**

三. 表达式和语句

1. 表达式：由运算符和运算对象组成（不带分号）。
2. 语句：一条语句相当于一个完整的计算机指令；在C中，大部分语句都以分号结尾。（声明不是语句，如int abc;不是语句，这与C++不同）。
3. 副作用：副作用是指对数据对象或文件的修改，比如使用赋值表达式和自增运算符就是利用其副作用。
4. 序列点：序列点是程序执行的点，在该点上，所有的副作用都在进入下一步之前发生（其实就是判断语句什么时候结束）主要的序列点：①分号；②完整的表达式：指

的是这个表达式不是其他任何表达式的子表达式（主要用于判断循环中条件结束，没什么好说的，只不过是一个噱头而已。）

5. 复合语句：一种特殊的语句，是C的六种语句之一，不用在}后面加；。

四. 类型转换

1. 类型转换规则：

①在表达式中，char和short类型都会被自动转化为int;

②作为函数参数传递时，不仅char和short被转化为int,连float都要转化为double;
有**原型**除外。③

涉及两种类型的运算，一般会转化到更高级别（升级），简单地来说，就是上限越大，级别越高。

④在**赋值表达式**中，右侧的结果最终会被转化为左侧的类型（可能出现降级）；

2. 强制类型转换：见上；

五. 函数初涉

本部分内容会在后面详讲，故这里也不会具体展开，稍微说一下重要的点：实参：argument，形参：parameter；