

# 第六讲：如何将算法告诉计算机

姓名： 丁保荣      学号： 171860509

2017 年 11 月 6 日

请独立完成作业，不得抄袭。  
若参考了其它资料，请给出引用。  
鼓励讨论，但需独立书写解题过程。

## 第一部分 作业

### 题目

写出你现在用的 C++ 语言的算术表达式的完整严格的文法

**注：** C++ 语言运算符来自 C++ Primer Plus(第六版) 附录 D 运算符优先级  
英语翻译水平较渣，还请见谅

### 解答：

```
<unary-operator> ::= !|+|-|++|--|&|*  
<binary-operator> ::= *|/|+|-  
<scope resolution operator> ::= <namespace> :: <statement>;  
<direct member operator> ::= <variable> . <variable>  
<array-index> ::= <array-variable> [<value>];  
<stepup> ::= <variable> ++ [;] | ++ <variable> [;]  
<stepdown> ::= <variable> -- [;] | -- <variable> [;]  
<negation> ::= !<expression>  
<positive-operator> ::= +<value>  
<negative-operator> ::= -<value>  
<address-access> ::= &<variable>  
<value-access> ::= *<pointer>  
<multiply-operator> ::= <value> * <value>  
<division-operator> ::= <value> / <value>  
<additive-operator> ::= <value> + <value>  
<subtraction-operator> ::= <value> - <value>  
<less-than-expression> ::= <value> < <value>
```

$\langle \text{greater-than-expression} \rangle ::= \langle \text{value} \rangle > \langle \text{value} \rangle$   
 $\langle \text{equality-expression} \rangle ::= \langle \text{value} \rangle == \langle \text{value} \rangle$   
 $\langle \text{not-equality-expression} \rangle ::= \langle \text{value} \rangle != \langle \text{value} \rangle$   
 $\langle \text{and-operator} \rangle ::= \langle \text{expression} \rangle \&\& \langle \text{expression} \rangle$   
 $\langle \text{or-expression} \rangle ::= \langle \text{expression} \rangle || \langle \text{expression} \rangle$   
 $\langle \text{assignment-operator} \rangle ::= = | * = | / = | \% = | + = | - = | \& = | =$   
 $\langle \text{combination-operator} \rangle ::= \langle \text{expression} \rangle , \langle \text{expression} \rangle$

---

## 第二部分 订正

### 题目 (DH:2.5)

Show how to perform the following simulations of some control constructs by others. The sequencing construct “and-then” is implicitly available for all the simulations. You may introduce and use new variables and labels if necessary.

- (a) Simulate a “for-do” loop by a “while-do” loop.
- (b) Simulate the “if-then” and “if-then-else” statements by “while-do” loops.
- (c) Simulate a “while-do” loop by “if-then” and “goto” statements.
- (d) Simulate a “while-do” loop by a “repeat-until” loop and “if-then” statements.

**错因分析：** 没有退出 while 的条件，会造成死循环

**订正：**

```

(b)
if (A) then
{
do B;
}
else
{
do C;
}
i=0;
while( (A) && (i==0))
{
do B;
i++;
}
while ( (not A) && (i==0) )
{

```

```
do C;  
i++;  
}
```

---

## 第三部分 反馈

你可以写：

- 对课程及教师的建议与意见
- 教材中不理解的内容
- 希望深入了解的内容
- 等