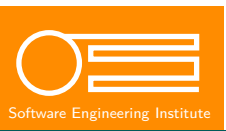
Types, Variables, Operators and Expressions (II)

Lecture 03

Min Zhang

zhangmin@sei.ecnu.edu.cn

2020.10.12



10:00-11:40, Monday, Room 319

Software Engineering Institute, East China Normal University

Review

Last lecture:

1. Data representation in computer
2. Types
3. Constants
4. Variables
5. Operators
6. Type conversion

Today:

1. Assignment operator
2. ++,--
3. Privilege of operators
4. Bit operators
5. Conditional expression

An exercise

```c
short int a='\xFF';
unsigned short int b='\xFF';
char c='\x30';
int d=031;
int e='\031';
printf("a=%d\n",a);
printf("b=%d\n",b);
printf("c=%c\n",c);
printf("c=%d\n",c);
printf("d=%d\n",d);
printf("e=%d\n",e);
```

Output:

```
a=____
b=____
c=____
c=____
d=____
e=____
```

An exercise

```c
short int a='\xFF';
unsigned short int b='\xFF';
char c='\x30';
int d=031;
int e='\031';
printf("a=%d\n",a);
printf("b=%d\n",b);
printf("c=%c\n",c);
printf("c=%d\n",c);
printf("d=%d\n",d);
printf("e=%d\n",e);
```

Output:

```
a=-1
b=65535
c=0
c=48
d=25
e=25
```

Type conversion

- Automatic conversion: from small-size type to big-size type

  Example: `5/2.0`, result: `2.5`. Here, `5` $\rightarrow$ `5.0`

- Compulsory Type Conversion: to force the conversion from a type to another one

  Example:

```
float pi=3.1415926;
int pi2 = (int) pi;
```

  The value of `pi2`: 3

Assignment = and assignment operator: op=

- Assignment =

```
1 int i=1;
2 i = i+2;
```

- Assignment operator op=

  op can be +, -, *, /, <<, >>, &, ^, |

```
1 int i=1;
2 i += 2; // same as the above code
```

Quiz

`x *= y+1; // x=2, y=3`

The value of x is 7 or 8?

Assignment = and assignment operator: op=

- Assignment =

```
1 int i=1;
2 i = i+2;
```

- Assignment operator op=

  op can be +, -, *, /, <<, >>, &, ^, |

```
1 int i=1;
2 i += 2; // same as the above code
```

Quiz

`x *= y+1; // x=2, y=3`

The value of x is 7 or 8? Answer: 8.

Remember: $expr_1\ op = expr_2 \triangleq expr_1 = (expr_1)\ op\ (expr_2)$

Value of expression

Every expression has a value

1. The value of 3+5: 8
2. The value of `x=3+5`: 8
3. The value of `x=(y=3)+5`: 8 (the value of `y` is 3)
4. The value of `(c=getchar()) != EOF`: 0 or 1
   `EOF` is a predefined symbol with value -1.
   `getchar()` is a function to read a character from keyboard.

Remarks

Remember:

The result of logical operation is only 1 or 0

Example: `int i='0'-48==0;`, value of `i`: 1

The privilege of logical operators is lower than arithmetic ones

Example: `int i='0'-48==0+1;`, value of `i`: 0

Operator ++ and --

++: Assume `int i=3;`

- `i++;` Result: The value of `i` is 4, the value of `i++` is 3 (先用再加)
- `++i;` Result: The value of `i` is 4, the value of `++i` is 4 (先加再用)

Quiz

```
int j=(i++)+(++i); // i=3
```

What is the value of `j`? Test by yourself!

Operator ++ and --

--: Assume `int i=3;`

- `i--;` Result: The value of `i` is 2, the value of `i--` is 3 (先用再减)
- `--i;` Result: The value of `i` is 2, the value of `--i` is 2 (先减再用)

Quiz

```
int j=(i--)+(--i); // i=3
```

What is the value of `j`? Test by yourself!

Bitwise operators

- Bitwise AND: &, `n=n&0177` // set to 0 except the rightmost 7 bits
- Bitwise inclusive OR: |, `n=n|0177` // the rightmost 7 bits are set to 1
- Bitwise exclusive OR: ^, `n=n^0177` // 0^0=0,1^1=0, otherwise 1
- Left shift: <<, `n=n<<2` // times by 4
- Right shift: >>, `n=n>>2` // divided by 4
- One's complement :~, `n=~n` ~0=1 and ~1=0

An example of bit operators

给定一个无符号整数 x，从该整数的二进制数中取出从第 p 位开始的 n 位二进制数，并计算该二进制数的值。

例如：假设 $x = 01010101$, $p = 4$, $n = 3$, 则红色部分 $010\color{red}{101}01$ 即是要求的结果。

得到101的方法：

- 第一步：x 向右移动两位，得到 00010101
- 第二部：与00000111进行与操作，除了后三位，其余全部变成 0

Example (Get n bits of x that begins at p)

```
x >> (p-n+1) & ~(~0 << n)
```

Conditional expression

```
expr1 ? expr2 : expr3
```

The value of `expr1 ? expr2 : expr3`

- If `expr1` is true, then `expr2`
- Otherwise, `expr3`

Example (Maximal value)

```
int max = (a >= b) ? a : b;
```

Precedence and order of evaluation

`() [] -> .`	从左至右
`! ~ ++ -- + - * (type) sizeof`	从右至左
`* / %`	从左至右
`+ -`	从左至右
`<< >>`	从左至右
`< <= > >=`	从左至右
`== !=`	从左至右
`&`	从左至右
`^`	从左至右
`|`	从左至右
`&&`	从左至右
`||`	从左至右
`?:`	从左至右
`= += -= *= /= %= &= ^= |= <<= >>=`	从右至左
`,`	从右至左

注：一元运算符+、-、&与*比相应的二元运算符+、-、&与*的优先级高。

Warning

Textbook, p.52, last paragraph

C, like most languages, does not specify the oder in which the operands of an operator are evaluated.

Example

```
int j=(i=4)+(i=i-1);                    //Assume i=3 before this statement
```

The values of `j` and `i` are

A `j=6, i=4`

B `j=7, i=4`

C `j=7, i=3`

D `j=6, i=3`

Answer: A,C

An example: condition of a leap year!

Given a year, write a program to check if it is a leap year.

Condition of a leap year: 四年一闰；百年不闰，四百年再闰

- 被 4 整除，且不被 100 整除，或，
- 被 400 整除

Let y be an int variable for the year

```
(y%4 == 0 && y%100 != 0) || (y%400 == 0)
```