

Java를 알고 C배우기

컴퓨터프로그래밍3

week 2-5 연산자

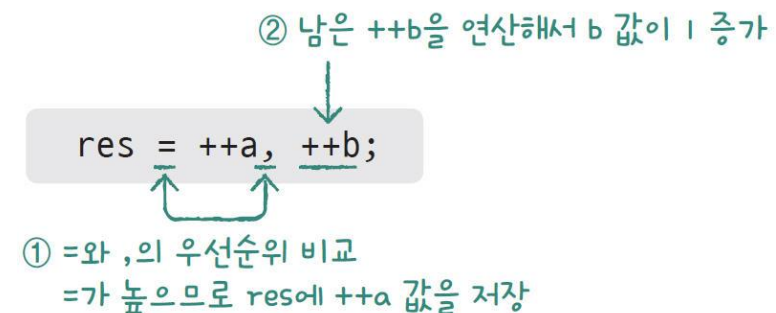
2022.1학기
충남대 조은선

C의 연산자

- ▶ 산술연산자 : + - * / % ++ --
- ▶ 관계연산자 : == > < >= <=
- ▶ 논리연산자 : && || !
- ▶ 비트연산자 : << >> & | ~
- ▶ 지정연산자 : = += -= *= /= %=
- ▶ 3항연산자 : ? :
 - ▶ + - 는 unary와 binary연산자가 별도로 존재
 - ▶ ++, -- 는 전위와 후위가 존재

연산자 우선순위와 연산 방향

종류	우선순위	연산자(괄호의 숫자는 우선순위)	연산 방향
1차 연산자	1	() [] . ->	→
단항 연산자	2	- ++ -- ~ ! * & sizeof (type)	←
산술 연산자	3	* / %	→
	4	+ -	
비트 이동 연산자	5	<< >>	
관계 연산자	6	< <= > >=	
동등 연산자	7	== !=	
비트 논리 연산자	8	&	→
	9	^	
	10		
논리 연산자	11	&&	
	12		
조건 연산자	13	?:	←
대입 연산자	14	= += -= *= /= %= &= ^= = <<= >>=	
coma 연산자	15	,	



C의 특징

- ▶ **Boolean 데이터 타입이 별도로 없음**

- ▶ 논리 연산의 결과는 숫자 1(참) 또는 0(거짓)이 된다
- ▶ 관계 연산의 결과도 숫자 1(참) 또는 0(거짓)이 된다

- ▶ 문제점

- ▶ $a > b > c$ 와 같은 식도 허용된다
 - ▶ $(a > b) > c$ 로 계산하므로, $a > b$ 가 참이면 $1 > c$ 과 의미가 동일
- ▶ `if (a = 0)`과 같은 식도 허용된다
 - ▶ a에 0을 지정하고, 지정문의 결과값이 r-value인 0이므로, 결과가 거짓이 되는 식임
 - ▶ 많이 쓰이는 것: `while (ch = getchar()) { ...}`

비트 연산자

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a = 10;           // 비트열 00000000 00000000 00000000 00001010
06     int b = 12;           // 비트열 00000000 00000000 00000000 00001100
07
08     printf("a & b : %d\n", a & b);
09     printf("a ^ b : %d\n", a ^ b);
10     printf("a | b : %d\n", a | b);
11     printf("~a : %d\n", ~a);
12     printf("a << 1 : %d\n", a << 1);
13     printf("a >> 2 : %d\n", a >> 2);
14
15     return 0;
16 }
```

실행결과

```
a & b : 8
a ^ b : 6
a | b : 14
~a : -11
a << 1 : 20
a >> 2 : 2
```

비트별 논리 합, 곱, 배타적 논리합, 부정연산자

▶ 비트별 논리곱 연산자(&)

```
00000000 00000000 00000000 00001010 (a = 10)
&) 00000000 00000000 00000000 00001100 (b = 12)
-----
00000000 00000000 00000000 00001000 (a & b = 8)
```

▶ 비트별 논리합 연산자(|)

```
00000000 00000000 00000000 00001010 (a = 10)
|) 00000000 00000000 00000000 00001100 (b = 12)
-----
00000000 00000000 00000000 00001110 (a | b = 14)
```

▶ 비트별 배타적 논리합 연산자(^)

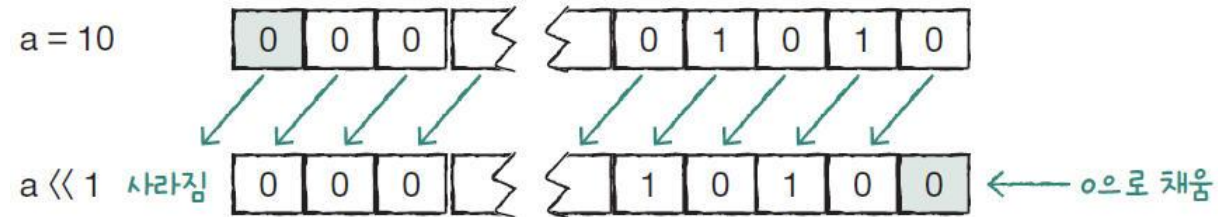
```
00000000 00000000 00000000 00001010 (a = 10)
^) 00000000 00000000 00000000 00001100 (b = 12)
-----
00000000 00000000 00000000 00000110 (a ^ b = 6)
```

▶ 비트별 부정 연산자(~)

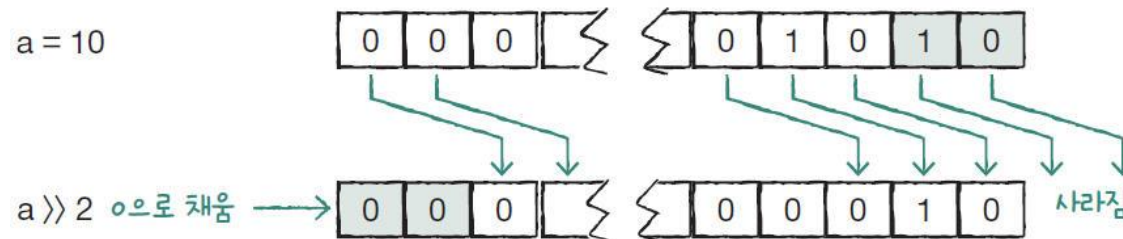
```
~) 00000000 00000000 00000000 00001010 (a = 10)
-----
11111111 11111111 11111111 11110101 (~a = -11)
```

Shift 연산자

▶ 비트별 왼쪽 이동 연산자(<<)



▶ 비트별 오른쪽 이동 연산자(>>)



Q) 사라진 비트 값으로 새 비트 자리에 채우려면?

복합 대입 연산자

복합대입 연산식	동일한 연산식	복합대입 연산식	동일한 연산식
$a += b$	$a = a + b$	$a \&= 2$	$a = a \& 2$
$a -= b$	$a = a - b$	$a \wedge= 2$	$a = a \wedge 2$
$a *= b$	$a = a * b$	$a = 2$	$a = a 2$
$a /= b$	$a = a / b$	$a \ll= 2$	$a = a \ll 2$
$a \% = b$	$a = a \% b$	$a \gg= 2$	$a = a \gg 2$

그 밖의 여러가지 연산자

연산자	연산식 예	결괏값
형 변환 연산자	<code>res = (int)10.7;</code>	res 값은 10
sizeof 연산자	<code>res = sizeof(double);</code>	res 값은 8
복합대입 연산자	<code>a += 10;</code>	a의 값을 10 증가
coma 연산자	<code>res = (a , b);</code>	res에 b 값 저장
조건 연산자	<code>res = (a > b) ? a : b;</code>	a가 b보다 크면 res 값은 a 작거나 같으면 res 값은 b
비트 연산자	<code>a & b; ~a; a << b;</code>	a와 b의 비트 상태에 따라 결괏값이 다름

Quiz

▶ 다음 중 사실과 가장 거리가 먼 것은?

- (1) C의 &&과 &는 의미가 다르다.
- (2) C의 참 거짓값은 Java와 데이터 타입이 다르다.
- (3) C에서 $a < b < c$ 와 같은 연산식을 컴파일하면 오류가 난다.
- (4) C의 sizeof는 인자로 받은 자료형의 크기를 내어준다.
- (5) C에서 $x = 0$ 과 같은 지정문을 if 의 조건절에 사용하면 컴파일 오류가 난다.