1. 연산자 우선순위(operator precedence)란

수식에서 어느 연산을 먼저 계산할 것인가를 결정하는 규칙

서로 다른 우선순위를 갖는 연산자에 대해선 우선순위가 높은 연산자가 먼저,

같은 우선순위를 가지는 연산자가 둘 이상 있을 때에는 결합 순서에 따라 실행 순서가 결정된다.

1. 연산자 우선순위 표

**연산자 우선순위 표**

|  |  |  |  |
| --- | --- | --- | --- |
| **우선순위** | **연산자** | **설명** | **결합 법칙(방향)** |
| 1순위 | x++ | 증가 연산자(뒤, 후위) | → |
| x-- | 감소 연산자(뒤, 후위) |
| ( ) | 함수 호출 |
| [ ] | 배열 첨자 |
| . | 구조체/공용체 멤버 접근 |
| -> | 포인터로 구조체/공용체 멤버 접근 |
| (자료형){값} | 복합 리터럴 |
| 2순위 | ++x | 증가 연산자(앞, 전위) | ← |
| --x | 감소 연산자(앞, 전위) |
| +x | 단항 덧셈 |
| -x | 단항 뺼셈 |
| ! | 논리 NOT |
| ~ | 비트 NOT |
| (자료형) | 자료형 캐스팅(자료형 변환) |
| \*x | 포인터 x역참조 |
| &x | x의 주소 |
| sizeof | 자료형의 크기 |
| 3순위 | \* | 곱셈 | → |
| . | 나눗셈 |
| % | 나머지 |
| 4순위 | + | 덧셈 | → |
| - | 뺄셈 |
| 5순위 | << | 왼쪽으로 비트를 시프트 | → |
| >> | 오른쪽으로 비트를 시프트 |
| 6순위 | < | 작음(논리) | → |
| <= | 작거나 같음(논리) |
| > | 큼(논리) |
| >= | 크거나 같음(논리) |
| 7순위 | == | 같음 | → |
| != | 다름 |
| 8순위 | & | AND(비트) | → |
| 9순위 | ^ | XOR(비트) | → |
| 10순위 | | | OR(비트) | → |
| 11순위 | && | AND(논리) | → |
| 12순위 | || | OR(논리) | → |
| 13순위 | ? : | 삼항 연산자 | ← |
| 14순위 | = | 할당 | ← |
| += | 덧셈 후 할당 |
| -= | 뺼셈 후 할당 |
| \*= | 곱셈 후 할당 |
| /= | 나눗셈 후 할당 |
| %= | 나머지 연산 후 할당 |
| <<= | 비트를 왼쪽으로 시프트 한 후 할당 |
| >>= | 비트를 오른쪽으로 시프트한 후 할당 |
| &= | 비트 AND 연산 후 할당 |
| ^= | 비트 XOR 연산 후 할당 |
| |= | 비트 OR연산 후 할당 |
| 15순위 | , | 쉼표 연산자 | → |

1. 참조 : <https://coding-factory.tistory.com/635>
2. 참조2 : https://blog.naver.com/huffman\_/222974844309