

## 3 장 연산자

---

### □ 개념 확인 학습

---

1. 연산에서 강제적으로 형변환이 필요한 경우 어떻게 처리해야 하나요?
2. 연산에서 자동으로 형변환이 일어나는 경우 자료형들의 우선순위가 어떻게 되나요?
3. 단항 연산자가 대입연산자와 함께 사용되는 경우 전위형( $b=++a;$ )과 후위형( $b=a++;$ )의 연산 결과는 어떻게 되나요?
4. 숫자로 표현되는 모든 자료형에서는 나머지 연산자( $\%$ )의 사용이 가능한가요?
5. 음의 정수값이 저장되어있는 변수에서 이동 연산자( $>>$ ,  $<<$ )를 사용할 경우 부호비트는 어떻게 처리되나요?
6. 비교 연산자( $==$ ,  $!=$  등)의 결과는 어떻게 처리되나요?
7. 비트 연산자  $\&$ 와 논리 연산자  $\&\&$ 의 다른 점은 무엇인가요?

8. C 언어에서 참과 거짓은 어떻게 구별되나요?
9. 조건 연산자(?)의 사용법을 설명하세요.
10.  $a \leq 3$  식의 의미는 무엇인가요?
11. 대입연산자와 괄호로 묶이지 않은 나열 연산식이 사용된 경우에 왼쪽의 변수에 대입되는 결과 값은 무엇인가요?

---

☐ 적용 확인 학습

---

1. 다음 서술 내용이 바르면 O, 그렇지 않으면 X를 표시 하세요.
- A. 정수형 변수 in을 강제로 double형으로 바꿀 때에는 `double(in);` 으로 형변환 한다. ( )
- B. `b = a++;` 을 수행한 후 b의 값은 a가 1증가하기 전의 값이다. ( )
- C. 단항 연산자 ‘!’ 와 ‘~’ 는 같은 의미이다. ( )

- D. 나머지 연산자 ‘%’의 피연산자는 반드시 정수여야 한다. ( )
- E. 나누기 연산자 ‘/’를 사용하여 정수나누기 정수를 했을 경우의 결과 값은 몫이 된다. ( )
- F.  $b = a \gg 3;$ 의 수행 결과로 b에는 a를 세 번 곱한 값이 저장된다. ( )
- G. 비교 연산자 ‘!=’은 두 피연산자의 값이 같을 경우 1을 반환한다. ( )
- H. 비트 XOR 연산자 ‘^’은 두 피연산자의 비트가 같으면 1을 반환한다. ( )
- I. 대입 연산식  $a \mathrel{+=} b;$ 는  $a = a + b;$ 이다. ( )
- J. 조건 연산식  $b ? c : d;$ 에서 b가 0이면 c가 수행된다. ( )
- K. 조건 연산식  $a = b ? c : d;$ 에서 b가 0이면 c의 수행 결과가 a에 대입된다. ( )
- L. 대입연산자와 괄호로 묶인 나열 연산식이 사용된 경우에는, 나열 연산식의 가장 오른쪽의 결과 값이 왼쪽의 변수에 대입된다. ( )

2. 다음에 서술된 문장에 어울리는 프로그램 코드를 완성해 보세요.

- A. 정수형 변수 num을 double형으로 형변환 한다.
- B. 문자형 변수 a에 저장되어 있는 값을 1감소시켜 b에 저장한다.

C. 정수형 변수 a, b, c 가 존재할 때, a를 b로 나눈 나머지를 c에 대입한다.

D. 정수형 변수 a, b가 존재할 때 이동 연산자를 사용하여 a에 8을 곱한 값을 b에 저장한다.

E. 정수형 변수 a, b, c가 존재할 때, 조건 연산자를 사용하여 a, b가 같으면 c에 100을 그렇지 않으면 c에 50을 대입한다. 이동 연산자를 사용하여 a에 8을 곱한 값을 b에 저장한다.

F. 실수형 변수 a, b가 존재할 때, a 보다 작은 최대 정수를 b에 저장한다.

3. 변수 값이 각각 a가 3일 때 다음 연산식의 결과 d를 쓰세요.

$d = (a++, b = a * 2, c = b - 2);$

$d = a++, b = a * 2, c = b - 2;$

---

☐ 응용 프로그래밍

---

>> 응용 프로그래밍은 다음과 같은 구조로 작성합니다. 번호가 늘어 갈 경우 번호에 해당하는 함수를 만들어 사용합니다.

```
#include <stdio.h>

void ex_02(void) {
    //응용 프로그래밍 2번은 ex_02() 내부에 작성합니다
}
void ex_03(void) {
    //printf("응용 프로그래밍 3번은 ex_03() 내부에 작성합니다.
}
int main() {
    ex_02();
    ex_03();
    return 0;
}
```

1. 커피 자판기에 돈을 넣으면 거스름돈을 내주는 시스템을 만들려고 합니다. 커피는 1200원이고 자판기는 10000원 까지 받을 수 있습니다. 거스름돈을 카운트하는 프로그램을 작성하세요. 단, 화폐단위는 5000원, 1000원, 500원, 100원 네 가지이며, 가능한 큰 화폐단위로 지불합니다. 예를 들어 10000원을 입력하고 커피 두 잔을 산다면 5000원권 1개, 1000원권 2개, 500원 동전 1개, 100원 동전 1개라고 출력하시면 됩니다.

---

```
#include <stdio.h>
int main(void)
{
    int x;

    printf("10000이하의 금액을 입력해 주세요 : ");
    ...
}
```

---

10000이하의 금액을 입력해 주세요 : 10000  
커피 몇 잔 드릴까요 : 2

거스름 돈 입니다.  
5000원권 1개  
1000원권 2개  
500원 동전 1개  
100원 동전 1개

2. 두 정수를 입력 받아 큰 수를 작은 수로 나눈 몫과 나머지를 각각 출력하는 프로그램을 작성하세요. (조건 연산자 사용)

정수 두 개를 입력하세요 : 20 30  
큰 수 = 30  
작은 수 = 20  
큰 수를 작은 수로 나눈 몫과 나머지 = 1, 10

3. getch()를 사용하여 문자를 입력 받아 알파벳이 아니면 'N', 알파벳이면 'Y'를 출력하는 프로그램을 작성하세요. (조건 연산자 사용)

문자를 입력하세요 : % -> N

문자를 입력하세요 : a -> Y

4. getch()를 사용하여 0~9까지의 숫자 2개를 입력 받아 두 숫자의 곱을 출력하세요. 단, 입력 받은 문자가 숫자가 아닌 경우에는 1로 처리하세요. 만약 7과 8을 입력 받았다면 56을 출력합니다. (조건 연산자 사용)

숫자를 입력하세요 : 7  
숫자를 입력하세요 : 8  
두 수의 곱은 = 56

숫자를 입력하세요 : 7  
숫자를 입력하세요 : &  
두 수의 곱은 = 7

5. 입력 받은 정수의 2배, 4배, 8배의 수를 출력하는 프로그램을 작성하세요.  
(곱하기 연산자 사용 금지)

정수를 입력하세요 : 2  
2배 = 4  
4배 = 8  
8배 = 16

정수를 입력하세요 : 3  
2배 = 6  
4배 = 12  
8배 = 24

6. 입력 받은 정수를 2로 나눈 몫이 10보다 크면 “OK” 그렇지 않으면  
“NOK” 를 출력하는 프로그램을 작성하세요. (나누기 연산자 사용 금지)

정수를 입력하세요 : 12  
NOK

정수를 입력하세요 : 30  
OK

7. 입력 받은 정수가 음수인지 양수인지 판별하세요. 음수는 부호비트가 1, 양수는 부호비트가 0입니다. (이동 연산자 사용)

정수를 입력하세요 : -1  
음수입니다.

정수를 입력하세요 : 1  
양수입니다.