

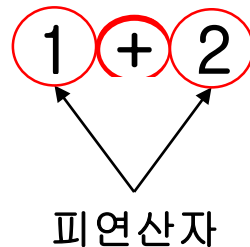
# JAVA 프로그램 실습

사칙 연산자

경북대학교  
소프트웨어융합과  
교수 배희호

# 용어 설명

- Operation(연산)
  - 정해진 규칙에 따라 Data를 처리하여 결과를 산출하는 것
- Operator(연산자)
  - 연산에 사용되는 표시나 기호
- Operand(피 연산자)
  - 연산자가 처리하는 Data
- Expression(수식)
  - 연산자와 피 연산자를 이용하여 연산의 과정을 기술한 것



# 용어 설명

## ■ Expression

- 어떤 값을 반환하는 표현 양식
- 연산자와 피 연산자로 구성

## ■ Statement

- Program의 논리적 기본 단위
- 이러한 문장들이 모여 Program을 구성

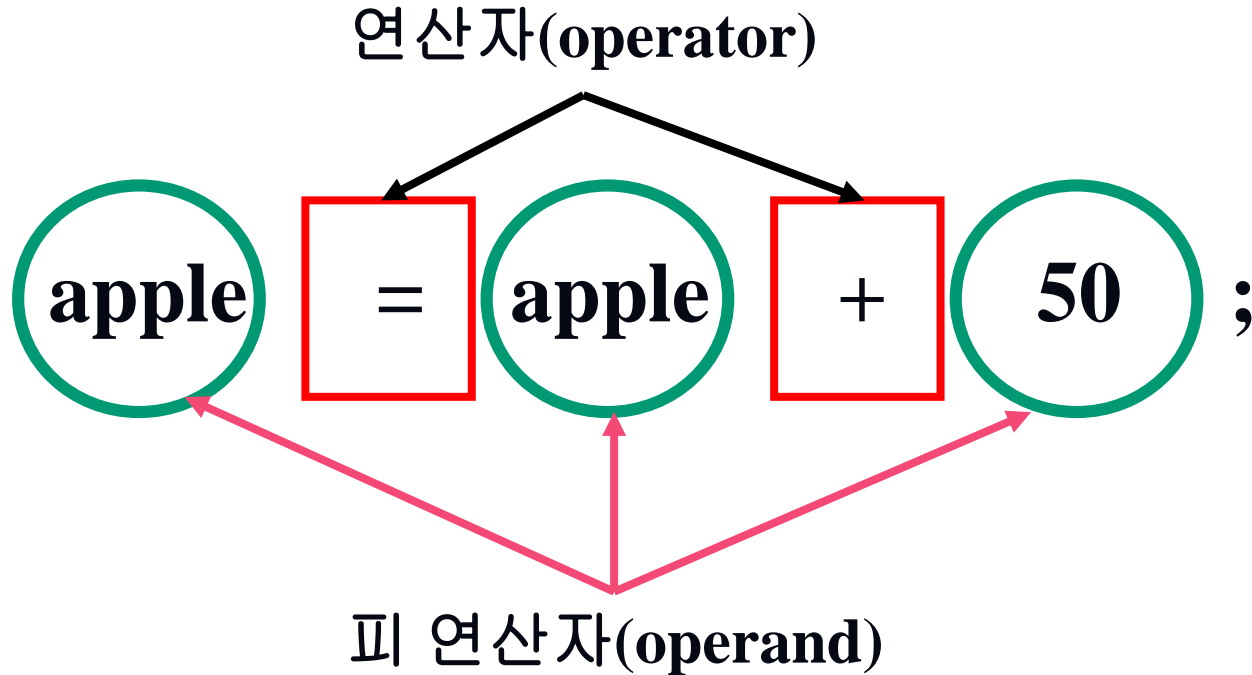
`1 + 2;`

`sum + 50; // sum에 50을 더한 값이 생성됨`

`test = sum + 50;`

# Operator 개요

- 문장에서 사용되는 수식은 연산자(Operator : +, -, \*, / 등)와 연산의 대상이 되는 피연산자(Operand : 숫자, 변수)로 구성



# Expression

- 상수, 변수, 함수, 연산자들의 결합으로 이루어진 JAVA의 표현 양식
- 모든 Expression은 반드시 결과 값을 갖음
- 모든 Expression은 소괄호( ( ) )에 의해 평가 순서를 표시함

8

result

a + b

a++

5.0 \* x - tan(9.0 / x)

a = b

/\* 값 : a 값과 b 값의 합 \*/

/\* 값 : 계산의 결과 \*/

/\* 값 : a에 지정되는 값 \*/

# Expression

수식(a = 2, b = 3)	결과값	의미
a		변수 하나로 구성된 표현식
10		상수 하나로 구성된 표현식
11.3		실수 상수로 구성된 표현식
a + 11.3		변수와 상수와 연산자로 구성된 표현식
a - b * 5.0		변수와 상수와 연산자로 구성된 표현식

$123$   $-123$   $v++$   
 $"is even\n"$   $'a'$   $n\%2 ? "odd" : "even"$   
 $a + 3.14$   $\max(x, y)$   
 $a > b$   $a * \max(x+1, y*2) / 2.0$   
 $a != 0 \ \&\& \ b*b - 4*a*c > 0$

# Expression의 형태

## ■ 1 차식

test

Identifier

7

Constant

(2 + a - 7)

Expression

## ■ 이항 수식(Binary Expressions)

test

첫 번째 항

+

이항 연산자

7

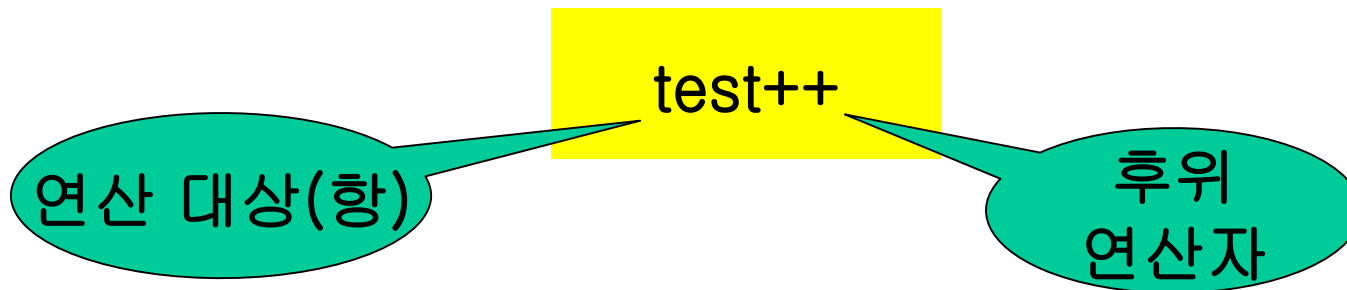
두 번째 항

# Expression의 형태

- 단항 수식(Unary Expressions)
  - 전위 수식(Prefix Expression)



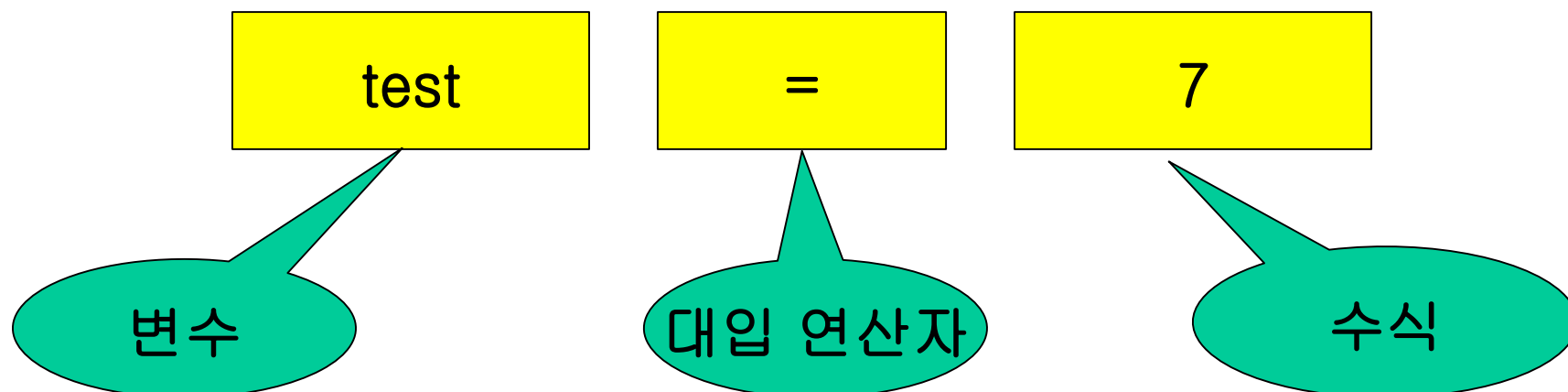
- 후위 수식(Postfix Expression)





# Expression의 형태

## ■ 대입식(Assignment Expression)



■ l-value : left – value

■ r-value : right – value

# Statement

- 형식적으로 Expression의 끝에 세미콜론(;)을 붙이면 됨
  - 세미콜론(;)은 Expression을 문장으로 만듦
- 문장은 글(Sentence)과 같이 행위(Action)를 가짐
- 문장은 JAVA Program 구성 단위인 클래스를 형성하는 기본 요소
- 문장은 의미 있는 문장, 의미 없는 문장으로 구분
- 의미 있는 문장은

| -value = 수식(expression);

- 각 문장은 다음 문장으로 넘어가기 전에 완전히 평가됨
- 공백 문장(null 문장)은 하나의 세미콜론(;)으로 작성됨
  - 공백 문장은 구문상으로는 문장이 필요하지만, 의미상으로 문장이 필요 없을 때 유용함

# Statement

## ■ 의미 없는 문장

■ 8;

■ result;

■ sqrt(7.333);

■ 5.0 \* x - tan(9.0 / x);

■ ;

/\* 값 : sqrt의 리턴 값 \*/

/\* 값 : 계산의 결과 \*/

/\* an empty statement \*/

## ■ 의미 있는 문장

■ a = b;

/\* 값 : a에 지정되는 값 \*/

■ sum = total + 10;

■ System.out.printf("%d\n", a);      /\* a method call \*/

# Operator 표기법

- Unary Operator는 접두(prefix) 연산자와 접미(postfix) 연산자로 사용

연산자 변수 (접두 연산자)  
변수 연산자 (접미 연산자)

- 2항 연산자는 삽입(infix) 표기법을 사용

수식1 연산자 수식2

- 3항 연산자

수식1 ? 수식2 : 수식3

# Operator 표기법

## ■ 피 연산자의 수에 따라 연산자 구분

연산자	예
단 항(unary)	$++a, a--, -a$
2 항(binary)	$a + b, a * b, a > b, a \&\& b$
3 항(ternary)	$\text{max} = (a > b) ? a : b;$

# Operator 표기법

- 단항 연산자
  - 피 연산자가 하나인 연산자
  - ‘++’, ‘—’, ‘!’, ‘~’ 부호를 의미하는 ‘+’와 ‘-’, 괄호(( )) 등이 있음
  - 접두(prefix) 연산자

## Operator 피 연산자 (접두 연산자)

- 예) ++a, --value, -test, !(flag), .....
- 접미(postfix) 연산자

## 피 연산자 Operator (접미 연산자)

- 예) b--, value--, .....

# Operator 표기법

## ■ 이항 연산자

- 2개 항 사이의 연산을 정의하는 연산자
- 삽입(infix) 표기법

피 연산자1 **operator** 피 연산자2

- 이항 연산자는 연산을 수행하기 전에 양항의 피 연산자의 Data Type을 일치시켜야 함
  - int 보다 크기가 작은 Type은 int로 변환
    - byte, char, short => int
  - 두 피 연산자 중 표현 범위가 큰 Type으로 형 변환 함
    - int VS float => float
    - float VS double => double

# Operator 표기법

## ■ 3항 연산자( ? : )

- 선택문의 if ~ else문을 축약해서 사용할 수 있는 연산자

수식1 ? 수식2 : 수식3

- 우선 수식1을 평가하여 true인지 false인지를 판별하여 true이면 수식2가 평가되고, false이면 수식3이 평가되어 그 결과가 수식1의 결과로 반환

$x > y$  가 참이면  $x$ 가 수식의 값이 된다.

$max\_value = (x > y) ? x : y;$

The diagram shows the expression  $max\_value = (x > y) ? x : y;$  with three colored shapes highlighting parts of it: a green hexagon under  $(x > y)$ , a blue hexagon under  $x$ , and a pink hexagon under  $y$ . Two red curved arrows originate from the question mark '?' and point to the green and pink shapes, indicating the conditional logic.

$x > y$  가 거짓이면  $y$ 가 수식의 값이 된다.



# JAVA의 연산자

구분		기능	연산자의 예
산술 연산자	이항 연산자	2개의 자료에 대해 산술 연산	+, -, *, /, %
	단항 연산자	1개의 자료에 대해 산술 연산	++, --
	대입 연산자	왼쪽 변수에 값을 대입	=, +=, -=, *=, /=, %=
관계 연산자		대소, 등가 관계를 비교	<, <=, >, >=, ==, !=
논리 연산자		복수개의 조건을 평가	&&,   , !
조건 연산자		조건 만족의 여부에 따라 별개의 값 계산	? 문장1 : 문장2
비트 연산자	비트 이동 연산자	비트 값을 좌우로 이동	<<, >>, >>>
	비트 논리 연산자	비트 값에 대해 논리 연산	&,  , ^, ~
기타 연산자	형변환 연산자	자료형 변환	(자료형)
	문자열 연산자	여러개의 문자열 결합	+

# Arithmetic Operator

- Arithmetic operator
  - 정수, 실수, 문자에 적용
  - Unary Operator(단항 연산자)
    - $+$ ,  $-$
  - Binary Operator(2항 연산자)
    - $+$ ,  $-$ ,  $*$ ,  $/$
    - $\%$  : 나머지 연산자

# Arithmetic Operator

- 기본적인 Arithmetic Operator 의미는 C 언어와 동일
- 더하기, 빼기, 곱하기, 나누기 등은 대부분 Programming 언어에서 사용되는 의미와 동일함

분류	연산자	연산식	예제		설명
			연산식	결과	
단항 연산자	+	+op	+7	7	양의 수임을 나타낸다
	-	-op	-7	-7	부호를 바꾼다
이항 연산자	+	op1 + op2	7 + 5	13	op1과 op2를 더하기
	-	op1 - op2	7 - 5	2	op1과 op2를 빼기
	*	op1 * op2	7 * 5	35	op1과 op2를 곱하기
	/	op1 / op2	7 / 5	1	op1과 op2를 나누기
	%	op1 % op2	7 % 5	2	op1을 op2로 나눈 나머지

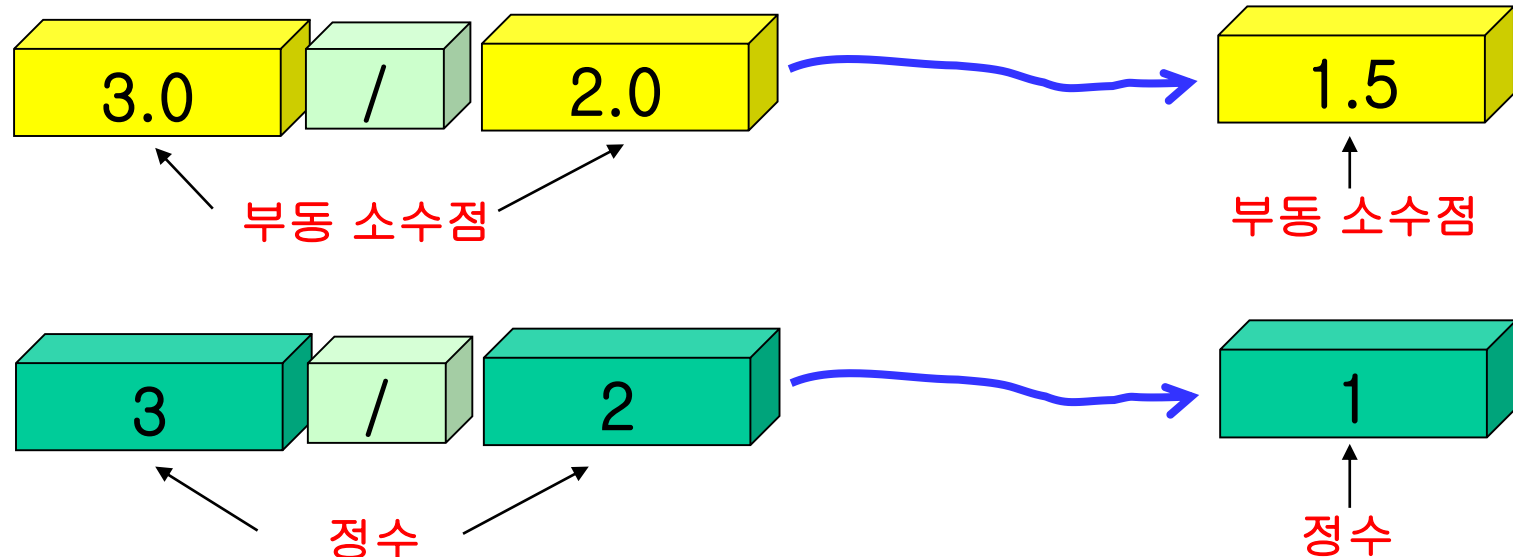
# Arithmetic Operator

- 산술 연산자의 결과는 피 연산자의 Data Type에 의해 결정

피 연산자	산술 연산자	피 연산자	결 과
정수형	+, -, *, /, %	정수형	정수형
정수형	+, -, *, /, %	실수형	실수형
실수형	+, -, *, /, %	정수형	실수형
실수형	+, -, *, /, %	실수형	실수형

# 나눗셈 연산자(/)

- '/'의 왼쪽 값을 오른쪽 값으로 나누는데 사용
- 정수형끼리의 나눗셈에서는 결과가 정수형으로 생성하고, 부동 소수점형끼리는 부동 소수점 값을 생성
- 정수형끼리의 나눗셈에서는 **소수점 이하는 버려짐**
- 정수와 부동 소수점을 섞어 연산할 때 연산 결과는 부동 소수점 (정수를 나눗셈을 하기 전에 부동 소수점으로 변환)



# 나눗셈 연산자(/)

- 정수형 / 정수형 : 몫(quotient)
  - $10 / 4 = 2$
- 실수형 / 실수형 : 나누기(divide)
  - $10.0 / 4.0 = 2.5$
- 나눗셈 연산자 주의(혼합형 나눗셈)

$$10 / 3 = 3$$

$$10 / 3.0 = 3.333333$$

$$10.0 / 3 = 3.333333$$

$$10.0 / 3.0 = 3.333333$$

$$\begin{array}{r} 1 \\ 5 \overline{) 7} \\ \underline{5} \\ 2 \end{array}$$

7/5

7%5

- 나눗셈을 하고 난 결과를 정수형 변수에다 넣어주게 되면  
그 결과는 정수형의 몫만이 기억

# 나머지 연산자(%)

- 나머지 연산자(modulus operator : %)는 나눗셈 연산의 나머지(remainder)를 반환
- 왼쪽 수를 오른쪽 수로 나누었을 때 나머지를 구하는데 사용
- 나머지 연산자는 정수뿐 아니라 실수에도 적용 가능
  - C 언어에서는 정수형에만 적용

```
int a = 12;
```

```
double b = 12.4;
```

```
System.out.println("result = " + a % 10);
```

```
System.out.println("result = " + b % 10);
```

```
result = 2
```

```
result = 2.4
```

실수형에도  
적용 가능

# 나머지 연산자(%)

- C/C++에서는 피 연산자로 float형 또는 double형 등과 같은 실수 자료형을 취할 수 없지만 JAVA에서는 가능 함

$$7 \% 3 = 1$$

$$7.2 \% 2.0 = 1.2$$

- 실수형 Data를 기반으로 하는 % 연산에서 Compile Error는 발생하지 않지만, 이는 수학적으로 의미가 없으므로 이러한 연산문을 작성하는 것은 주의를 요함
- 나머지 연산자의 부호는 왼쪽 항의 부호를 따름

$$10 \% 8 = 2$$

$$10 \% -8 = 2$$

$$-10 \% 8 = -2$$

$$-10 \% -8 = -2$$



# Arithmetic Operator 예제 1

■ 다음 수식을 Program의 연산 수식으로 표현해보자

번호	수식	Program 연산 수식 표현
예	$a + 3b$	$a + 3 * b$
1	$a(b - 3)$	
2	$c - ab$	
3	$b^2 - 4ac$	
4	$a + 3 / b$	
5	$a / b + c$	
6	$a + b / x - y$	

# Arithmetic Operator 예제 2

- 다음 Program의 출력 결과는 ?

```
public static void main(String[] args) {  
    System.out.println(10 % 8);  
    System.out.println(10 % -8);  
    System.out.println(-10 % 8);  
    System.out.println(-10 % -8);  
    System.out.println(10.5 % -8);  
    System.out.println(-10.5 % -8);  
}
```

2  
2  
-2  
-2  
2.5  
-2.5

나머지 연산자의 부호는 왼쪽 항의 부호를 따름  
부동 소수점 수에도 적용 가능

# Arithmetic Operator 예제 3

■ 다음 Program의 출력 결과는 ?

```
public static void main(String[] args) {  
    System.out.println(10 / 3);  
    System.out.println(10 / 3.0);  
    System.out.println(10.0 / 3);  
    System.out.println(10.0 / 3.0);  
    System.out.println(10 / 3.0f);  
    System.out.println(10.f / 3.0);  
}
```

```
3.3333333333333335  
3.3333333333333335  
3.3333333333333335  
3.3333333  
3.3333333333333335  
3.3333333333333335
```

정수와 부동 소수점을 섞어 연산할 때  
연산 결과는 부동 소수점

# Arithmetic Operator 예제 4

■ 다음 수식의 결과를 설명하여라

수 식	연산 과정	결 과
$2 + 4$ $2.5 + 6$ $4.0 / 5$ $10 * 4 / 4$ $10 / 4 * 4$ $10 / 4.0 * 4$ $5 / 4$	$2 + 4$	6

$x = 1.7;$       1 (만약 x가 정수형 변수라면)  
                  1.7 (x가 실수형 변수라면)

# Arithmetic Operator 예제 4

■ 다음 수식의 결과를 설명하여라

수 식	연산 과정	결 과
7 / 2		
7.0 / 2		
7 / 2.0		
7.0 / 2.0		
7 % 2		
7.5 % 2.0		
-9 % 3		

# Arithmetic Operator 예제 5

■ 산술 대입 연산자로 표현하여 보아라

번호	단순 대입 연산자	산술 대입 연산자
예	$i = i + 1;$	$i \text{ += } 1;$
1	$n = n * (k+1);$	
2	$d = d / 12;$	
3	$a = a + 2;$	
4	$\text{count} = \text{count} - 4;$	
5	$i = i + (j - k);$	

# Arithmetic Operator 예제 6

■ 다음 프로그램의 출력 결과는 ?

```
public static void main(String[] args) {  
    System.out.println(1 / 2 - 3);  
    System.out.println(2.0 + 1.5 * 2.0);  
    System.out.println(10 / 3 / 2);  
    System.out.println(2 * 5 / 2);  
}
```

-3  
5.0  
1  
5

# 단항 연산자(+, -)

- 단항 연산자로 쓰일 경우에는 상식적으로 알고 있는 Sign(부호)를 의미하는 연산자
- 부호 연산자(+, -) ‘+’는 피 연산자에 1을 곱하고, ‘-’는 피 연산자에 -1을 곱함

```
int a = -2;  
int b = +9;  
int c = -3;  
int d = -12;
```

a = -2, b = -9, c = 3, d = -12

해당 기억장소의 값에 부호를 붙여 연산이나 출력을 하는 의미이며 **기억장소 자체의 값이 부호가 바뀌지는 않음**

```
System.out.printf("a = %d, b = %d, ", a, -b);  
System.out.printf("c = %d, d = %d\n", -c, +d);
```



# 증가 감소 연산자(++ , --)

- 증감 연산자는 C, Perl 언어에서도 지원되는 일반적인 연산자이며, 역할도 동일함
- 증감 연산자의 특이점은 변수의 앞 또는 뒤에 위치함에 따라서 반환값(return value)이 달라짐

연산자	사용 예	의미
++	++test;	연산 전에 test의 값을 1 씩 증가
	test++;	연산 후에 test의 값을 1 씩 증가
	test = test + 1;	
--	--test;	연산 전에 test의 값을 1 씩 감소
	test--;	연산 후에 test의 값을 1 씩 감소
	test = test - 1;	

# 증가 감소 연산자(++ , --)

## ■ 전위형(prefix)

```
y = ++x;
```

```
x = x + 1;  
y = x;
```

## ■ 후위형(postfix)

```
y = x++;
```

```
y = x;  
x = x + 1;
```

1 증가 연산 문장	1 감소 연산 문장
count = count + 1; count += 1; count++;	count = count - 1; count -= 1; count--;

# 증가 감소 연산자(++ , --)

- 변수에만 적용가능 (수식이나 상수에는 적용 못함)

```
c = ++(a + 9);  
798++;
```

- 함수의 인자로 2번 이상 사용되는 매개변수에 사용 하지 말 것

```
num = 5;  
System.out.printf("%d, %d\n", num, num * num++);
```

5, 25

6, 25

(둘 중에 어느 것이 출력될지 보장할 수 없음)

# Arithmetic Operator 예제 7

■ 다음 Program의 결과를 설명하여라

문1)

```
int a = 10, b = 20, c;
```

```
a = 11, b = 20, c = 31
```

```
c = ++a + b;
```

```
System.out.printf("a = %d, b = %d, c = %d", a, b, c);
```

문2)

```
int a = 10, b = 20, c;
```

```
c = a++ + b++;
```

```
System.out.printf("a = %d, b = %d, c = %d", a, b, c);
```

# Arithmetic Operator 예제 7

문3)

```
int a, b, c;
```

```
a = 3, b = 4;
```

```
c = ++a + b++;
```

```
System.out.printf("a = %d, b = %d, c = %d", a, b, c);
```

문4)

```
int a, b, c;
```

```
a = 3, b = 4;
```

```
c = ++a + ++b;
```

```
System.out.printf("a = %d, b = %d, c = %d", a, b, c);
```



# Arithmetic Operator 예제 7

문5)

```
int a, b;
```

```
a = 3, b = 4;
```

```
System.out.printf("a = %d, b = %d", ++a, b++);
```

```
System.out.printf("\n a = %d, b = %d", a, b);
```

문6)

```
int a, b;
```

```
a = 3, b = 4;
```

```
System.out.printf("a = %d, b = %d", --a, b--);
```

```
System.out.printf("\n a = %d, b = %d", a, b);
```

# Arithmetic Operator 예제 8

- 다음 Code를 실행시키면 i, j, k 변수가 갖게 되는 값이 무엇인지 골라보세요.

```
int k;  
int i = 5;  
int j = 6;  
k = i++ + ++j;
```

- ① k=12, i=6, j=7
- ③ k=12, i=5, j=7

- ② k=11, i=5, j=7
- ④ k=11, i=6, j=7

## [해설]

i++ 연산에 의해 현재 i변수의 값 5와 ++j 연산에 의해 현재 j변수의 값에서 1 증가한 값 7을 더하면 k 변수는 12이다.  
그리고 나서 i변수의 값은 증가된 6의 값을 가지게 됨

# Arithmetic Operator 예제 9

- 다음 Program의 실행 결과를 설명하여라.

```
value = 1;  
result = (value++ * 5) + (value++ * 3);  
System.out.printf("%d, %d\n", value, result);
```

- value에 5를 곱하고 value에 1을 더한다
- value에 3을 곱하고 value에 1을 더한다
- 두 곱의 결과를 더한다

한 문장 내에 두 번 이상  
사용되는 변수에 대해서는  
사용하지 말 것



# Arithmetic Operator 예제 10

- 다음 Program램의 실행 결과를 설명하여라

```
int i = 1;  
int j = i++;  
if ((i > ++j) && (i++ == j)){  
    i += j;  
}
```

i = 2, j = 2

# Arithmetic Operator 예제 1

```
public static void main(String[] args) {  
    int a = 10, b = 20, c;  
    c = ++a + b;  
    System.out.printf("a = %d, b = %d, c = %d\n", a, b, c);  
  
    int a1 = 10, b1 = 20, c1;  
    c1 = a1++ + b1++;  
    System.out.printf("a1 = %d, b1 = %d, c1 = %d\n", a1, b1, c1);  
  
    int a2, b2, c2;  
    a2 = 3;  
    b2 = 4;  
    c2 = ++a2 + b2++;  
    System.out.printf("a2 = %d, b2 = %d, c2 = %d\n", a2, b2, c2);  
  
    int i = 5, j = 6;  
    int k = i++ + ++j;  
    System.out.printf("i = %d, j = %d, k = %d\n", i, j, k);  
}
```

```
a = 11, b = 20, c = 31  
a1 = 11, b1 = 21, c1 = 30  
a2 = 4, b2 = 5, c2 = 8  
i = 6, j = 7, k = 12
```

# Assignment Operator

- 대입 연산자는 왼쪽 항인 l-value(변수, 기억 장소)에 오른쪽 수식의 값을 기억시키는 이항 연산자
- 대입 연산자의 연산 기호는 '='(assign)임
- 서로 같다(equal)는 수학적 기호와 는 의미가 다름
- = 왼쪽은 변수이고 오른쪽은 변수의 값임

```
apple = 5;
```

-> 변수 apple에 5 란 정수값을 저장(대입)

- 대입 연산자는 오른쪽에서부터 왼쪽으로 평가

```
apple = banana = 5;
```

- 대입 연산자의 왼쪽 피 연산자는 상수가 아니고 반드시 변수 이어야 함

# Assignment Operator

- 대입 연산자 양변의 Data Type은 일치해야 함

```
int test = 100;  
float test = 34.5f;
```

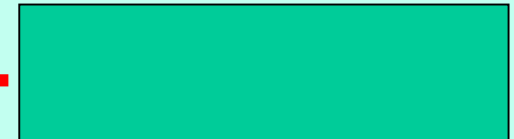
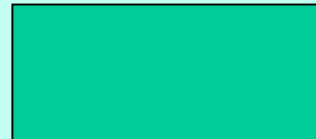


- 대입 연산자 양변의 Data Type 일치하지 않는 경우(오류)

```
long test = 100;  
double test = 34.5f;
```



```
int test = 40000L;  
float test = 34.5;
```



# Assignment Operator

- l-value와 r-value
  - 대입 연산자의 왼쪽(left-value)과 오른쪽(right-value)을 의미
  - l-value
    - 값이 저장되는 위치(주소, 참조)
  - r-value
    - 저장되는 값  
(수식, 변수, 상수, 포인터, 배열원소 등)

test = test + 1;

l-value

r-value

# Assignment Operator

## ■ l-value와 r-value

자료형	l-value	r-value
변수	변수의 주소	변수의 값
상수( 3.5)	없음	값 3.5
배열 “A[i]”	i 번째 원소 위치	원소 A[i]의 값

# 산술 대입 연산자

- C, C++, Perl 등의 언어에서도 산술 연산과 대입 연산을 결합하여 표현할 수 있으며, JAVA에서도 이를 지원하고 있음

연산자	사용법	의미
+=	op1 += op2	op1 = op1 + op2
-=	op1 -= op2	op1 = op1 - op2
*=	op1 *= op2	op1 = op1 * op2
/=	op1 /= op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2

# 산술 대입 연산자

- 산술 대입 연산자는 산술 연산자(+, -, \*, /, %)와 대입 연산자(=)가 결합된 이항 연산자

변수 **산술 연산자** = 수식2

변수 = 변수 산술 연산자 (수식2)

- 일반적으로 모든 이항 연산자들은 대입 연산자(=)와 결합하여 표현될 수 있음
- 예)

(1) num **+=** 10;

↔ num = num + 10;

(2) count **\*=** value + 2;

↔ count = count \* (value + 2);

- 산술 대입 연산자는 수식의 복잡성을 줄여 효율적 임



# 산술 대입 연산자

- 산술 연산과 대입 연산을 결합하여 표현하면, 코딩의 길이가 짧아짐
- 보다 효과적인 실행 Code를 발생시켜서 Program의 실행 빨라짐
- 가능한 대입 연산과 산술 연산을 결합하여 표현하는 것을 권장함

# 산술 연산자 예제 12

■ 아래 수식을 평가해보자

```
int a = 1, b = 2, c = 3, d = 4;
```

수식	동일한 수식		값
$a * b / c$	$(a * b) / c$		
$a * b \% c + 1$	$((a * b) \% c) + 1$		
$++a * b - c--$	$((++a) * b) - (c--)$		
$7 - -b * ++d$	$7 - ((-b) * (++d))$		
$a += b + c$	$a += (b + c)$	$a = (a + (b + c))$	
$b *= c = d + 5$	$b *= (c = (d + 5))$	$b = (b * (c = (d + 5)))$	

# 도입 예제

- 다음과 같이 나눗셈을 하는 Program을 작성해보자

첫 번째 수를 입력 (분자) : **12<enter>**

두 번째 수를 입력 (분모) : **7<enter>**

12를 7로 나누면 몫 = 1, 나머지 = 5 이다.

12를 7로 나누면 = 1.7 이다.



# 도입 예제

## ■ 문제 분석

첫 번째 수를 입력 (분자) : 12<enter>

두 번째 수를 입력 (분모) : 7<enter>

데이터

12를 7로 나누면 몫 = 1, 나머지 = 5 이다.

12를 7로 나누면 = 1.7 이다.

정보



# 도입 예제

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 숫자1(num1) : 정수형 (int)
- 숫자2(num2) : 정수형 (int)

## ■ 출력

- 나눗셈(div) : 정수형 (int), 실수형(float)
- 나머지(remainder) : 정수형 (int)

## ■ 계산 방법

- $div = num1 / num2$
- $Remainder = num1 \% num2$
- 분모가 0이면 오류 발생

# 도입 예제

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int num1, num2;  
    int div, remainder;  
    float result;  
  
    System.out.print(" 첫 번째 수를 입력 (분자) : ");  
    num1 = keyboard.nextInt();  
    System.out.print(" 두 번째 수를 입력 (분모) : ");  
    num2 = keyboard.nextInt();  
  
    div = num1 / num2;  
    remainder = num1 % num2;  
    result = (float) num1 / num2;
```

# 도입 예제

```
System.out.printf("\n %d를 %d로 나누면 몫 = %d, 나머지 = %d 이다.",  
    num1, num2, div, remainder);  
System.out.printf("\n %d를 %d로 나누면 = %f 이다.\n ",  
    num1, num2, result);  
}
```

# 실수 입력

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float num1, num2, num3;  
  
    System.out.print("세개의 실수를 입력 하세요 : ");  
    num1 = keyboard.nextFloat();  
    num2 = keyboard.nextFloat();  
    num3 = keyboard.nextFloat();  
  
    System.out.print("num1 = " + num1);  
    System.out.println("\nnum2 = " + num2);  
    System.out.printf("num3 = %f\n", num3);  
}
```



# 환율 계산

- 원화(W)를 달러(\$)로 계산 하는 Program을 작성하여라.

달러에 대한 원화 환율을 입력 : 928.78 <Enter>

원화 금액을 입력 : 1000000 <Enter>

원화 1,000,000원은 1,076.681204 달러입니다.

# 환율 계산

- 문제 분석
  - Data는 무엇이고, Information는 무엇인가?
- 입력
  - 달러 환율 - float
  - 원화 금액 - int
- 출력
  - 환전한 달러 금액 - float
- 계산 방법
  - / 연산자 사용

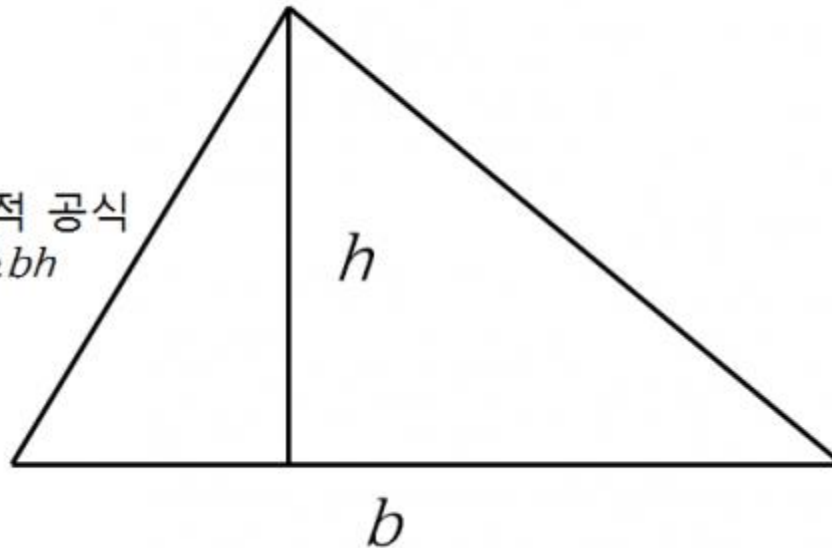
# 환율 계산

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float dollar;  
    int money;  
    float exchange;  
  
    System.out.print("달러에 대한 원화 환율을 입력 : ");  
    dollar = keyboard.nextFloat();  
    System.out.print("원화 금액을 입력 : ");  
    money = keyboard.nextInt();  
  
    exchange = money / dollar;  
  
    System.out.printf("원화 %,d원은 %,f 달러입니다.", money, exchange);  
}
```

# 삼각형 넓이

- 삼각형의 넓이(area)를 구하는 프로그램을 작성 하여라.

삼각형 면적 공식  
 $A = \frac{1}{2}bh$



# 삼각형 넓이

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 밑변(base) – int

- 높이(height) – int

- 출력

- 삼각형 면적(area) – float

- 계산 방법 (알고리즘)

- 삼각형의 넓이 = 밑변(base) x 높이(height) / 2

# 삼각형 넓이

## ■ 화면 설계

\*\*\*\* 삼각형의 넓이 구하기 \*\*\*\*

밑변 : 10

높이 : 3

넓이 : XX.XX <--- 소수 2자리까지 출력

# 삼각형 넓이

```
public static void main(String[] args) throws IOException {  
    BufferedReader br = new BufferedReader(new  
        InputStreamReader(System.in));  
    int bottom, height;  
  
    System.out.println("*** 삼각형의 넓이 구하기 ***");  
    System.out.print("밑변 : ");  
    bottom = Integer.parseInt(br.readLine());  
    System.out.print("높이 : ");  
    height = Integer.parseInt(br.readLine());  
  
    double area = (double)(bottom * height) / 2;  
  
    System.out.printf("넓이 : %.2f", area);  
}
```

# 키와 몸무게

- 킬로그램(kg) 단위의 몸무게를 파운드(pound)로, 센티미터(cm) 단위의 키를 피트(feet)로 바꾸는 Program을 작성하라.

당신의 키는 얼마입니까 ?(Cm) : 176.2 <enter>

당신의 몸무게는 얼마입니까 ?(Kg) : 72.6 <enter>

당신의 키는 5.781 피트(feet) 입니다.

당신의 몸무게는 160.056 파운드(pound)입니다.



# 키와 몸무게

- 문제 분석
  - Data는 무엇이고, Information는 무엇인가?
- 입력
  - 키 (height) – float
  - 몸무게 (weight) – float
- 출력
  - 피트 (feet) – float
  - 파운드 (pound) – float
- 계산 방법
  - 1피트(feet)는 30.48 센티미터(cm)로 하고
  - 1파운드(pound)는 0.453592 킬로그램(kg)으로

# 키와 몸무게

- 구현 (알고리즘)
  - 표준입력으로 킬로그램(kg) 단위의 몸무게, 센티미터(cm) 단위의 키를 입력
  - 입력 받은 자료를 각각 파운드(pound)와 피트(feet)로 변환하여 몸무게와 키를 출력
  - 출력 (문제의 Form 활용)

# 키와 몸무게

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float height, weight;  
    final float FEET = 30.48f;  
    final float POUND = 0.453592f;  
    float feet, pound;  
  
    System.out.print("당신의 키는 얼마입니까?(Cm) : ");  
    height = keyboard.nextFloat();  
    System.out.print("당신의 몸무게는 얼마입니까?(Kg) : ");  
    weight = keyboard.nextFloat();  
    feet = height / FEET;  
    pound = weight / POUND;  
    System.out.printf("당신의 키는 %.3f 피트(feet) 입니다.\n", feet);  
    System.out.printf("당신의 몸무게는 %.3f 파운드(pound)입니다.\n",  
                                                                pound);  
}
```

# 표준체중

- 키(height)를 입력 받아 표준체중(standard)을 구하는 Program을 작성하여라
  - 표준체중(kg) = [신장(cm)-100] x 0.9



# 표준체중

## ■ 문제분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 키(신장) : height – float

## ■ 출력

- 표준체중 : standard – float

## ■ 계산 방법

### ■ 표준체중

- 신장(키)을 기준으로 하였을 때 사망률이 가장 낮은 체중을 말함
- 표준체중(kg) = [신장(cm)-100] x 0.9

# 표준체중

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float height, standard;  
  
    System.out.print("당신의 키는 얼마입니까?(Cm) : ");  
    height = keyboard.nextFloat();  
  
    standard = (height - 100) * 0.9f;  
  
    System.out.printf("키 %.1f Cm의 표준체중은 %.2f Kg입니다.\n",  
        height, standard);  
}
```

# Mile을 Km로 변환

- 마일(mile)을 킬로미터(kilometer)로 변환하는 Program을 작성하여라.
  - 1 마일은 1.609 Km



# Mile을 Km로 변환

## ■ 문제 분석

- 데이터는 무엇이 있는가요 ? (입력)

  - 마일(mile) : int

- 정보는 무엇을 요구하나요 ? (출력)

  - 킬로미터(kilometer) : float

- 정보를 어떻게 구할 수 있을까 ? (알고리즘 (계산 방법))

  - 1 마일은 1.609 Km

## ■ 화면 설계



# Mile을 Km로 변환

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final float MILE = 1.609f;  
    int mile;  
    float km;  
  
    System.out.print("Mile 입력 : ");  
    mile = keyboard.nextInt();  
  
    km = mile * MILE;  
  
    System.out.printf(" %d Mile = %.2f Km\n", mile, km);  
}
```

# 원의 면적과 원주값 구하기

- 원의 반지름이 주어질 때 이 원의 면적과 원주값을 계산하여라.



# 원의 면적과 원주 값 구하기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 원의 반지름(radius) - int

- 원주율 상수  $\pi$ (계수)는 고정된 값 : 3.141592

## ■ 출력

- 원의 면적(area) - double

- 원주값(circum) - double

## ■ 알고리즘

- 원의 면적 =  $\pi(3.141592) \times (\text{반지름})^2$

- 원의 둘레 =  $2 \times \pi(3.141592) \times \text{반지름}$

- 고정된 값 : 상수

final double PI = 3.141592;

# 원의 면적과 원주 값 구하기

## ■ 가상 언어 표현

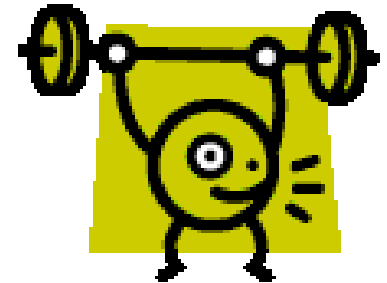
1. 원의 반지름을 입력 받는다
2. 원의 면적을 계산한다.

$$\text{원의 면적} = \pi * \text{반지름} * \text{반지름}$$

3. 원의 둘레를 계산한다.

$$\text{원의 둘레} = 2 * \pi * \text{반지름}$$

4. 원의 면적과 둘레를 출력한다.



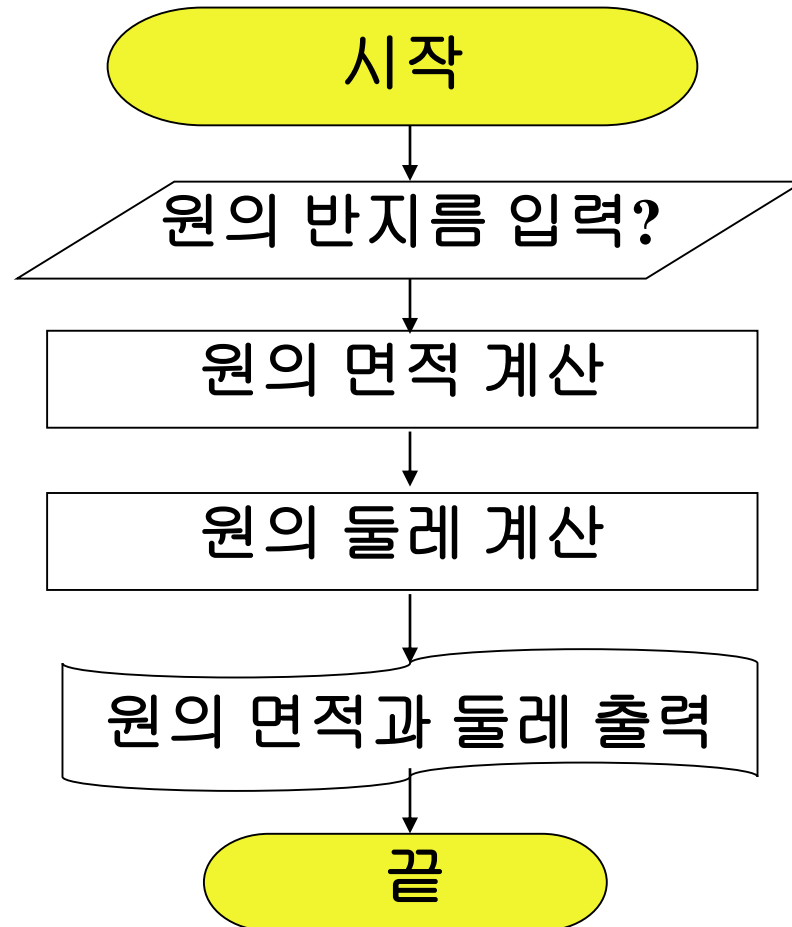
입력 (데이터)

계산

출력 (정보)

# 원의 면적과 원주 값 구하기

## ■ 가상 언어 표현



# 원의 면적과 원주 값 구하기

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int radius;          /* 입력 - 원의 반지름 */  
    final double PI = 3.141592;  
    double area;         /* 출력 - 원의 면적 */  
    double circum;       /* 출력 - 원의 둘레 */  
  
    System.out.printf("원의 반지름 입력 : ");  
  
    radius = keyboard.nextInt();  
    area = PI * radius * radius;  
    circum = 2 * PI * radius;  
  
    System.out.printf("반지름이 %d Cm인 원의 면적 : %.4f \n", radius, area);  
    System.out.printf("반지름이 %d Cm인 원의 둘레 : %.4f \n", radius, circum);  
}
```

# 원의 면적과 원주 값 구하기

## ■ 실행 결과 검사

- 주어진 숫자(반지름)을 가지고 직접 계산 후 실행 결과와 비교, 확인

원의 반지름 입력 : 5

반지름이 5 Cm인 원의 면적 :  $78.5398\text{cm}^2$

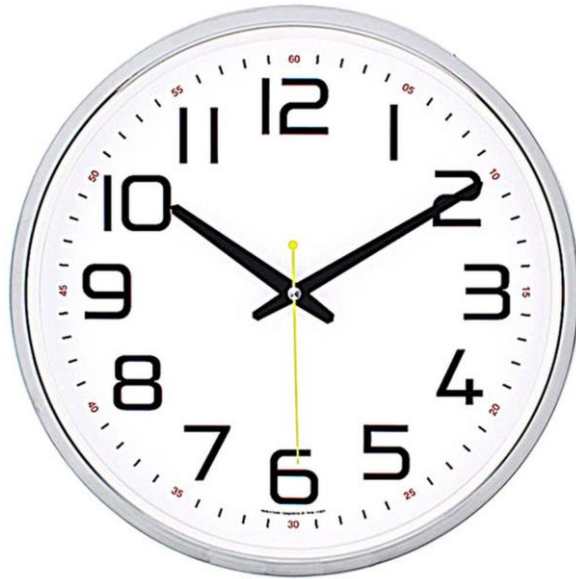
반지름이 5 Cm인 원의 둘레 :  $31.4159\text{cm}$



# 시간 환산

- 시간을 초 단위로 입력하여 몇 시간 몇 분 몇 초인지를 계산하는 Program을 작성하시오.

원하는 시간을 초단위로 입력 : 12000<enter>  
12000 초는 0일 3시간 20분 0초





# 시간 환산

- 문제 분석
  - Data는 무엇이고, Information는 무엇인가?
- 입력
  - 초 단위 시간(time) – int
- 출력
  - 일(day) – int
  - 시간(hour) – int
  - 분(minute) – int
  - 초(sec) – int
- 계산 방법
  - %, / 연산자 사용

# 시간 환산

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int BASE = 60;  
    final int DAY = 24;  
    int time;  
    int day, hour, minute, sec;  
  
    System.out.print("초단위 시간을 입력하세요 : ");  
    time = keyboard.nextInt();  
  
    sec = time;  
    minute = sec / BASE;  
    sec %= BASE;  
    hour = minute / BASE;  
    minute %= BASE;  
    day = hour / DAY;  
    hour %= DAY;
```

# 시간 환산

```
System.out.printf(
    "%,d seconds = %,d day %,d hour %,d minute %,d sec.\n",
    time, day, hour, minute, sec);
}
```

# Coffee 비용 계산

- 하루에 평균  $X$ 잔의 커피를 마신다. 평생(60년) 동안에는 몇 잔의 커피를 마시고, 커피 한잔에 평균  $Y$ 원하면 평생 얼마의 비용이 필요한지 계산하는 Program을 작성하여라.



# Coffee 비용 계산

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 하루에 마신 Coffee – int (coffee)
  - Coffee 가격 – int (price)
  - 평생 : 60년

- 출력

- 평생 마신 Coffee 양 – int cup
  - 평생 마신 Coffee 금액 – int money

- 계산 방법

- 사칙 연산

# Coffee 비용 계산

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    final int LIFE = 60;  
    int coffee;  
    int price;  
    int cup;  
    int money;  
  
    System.out.printf(" 하루에 몇잔의 Coffee를 마시는지요 ? ");  
    coffee = keyboard.nextInt();  
    System.out.printf(" Coffee 한잔의 가격은 ? ");  
    price = keyboard.nextInt();  
  
    cup = LIFE * 365 * coffee;  
    money = cup * price;  
}
```

# Coffee 비용 계산

```
System.out.printf("하루에 %d잔의 Coffee를 마신다\n", coffee);  
System.out.printf("평생(%d 년) 마신 Coffee는 %,d 잔 입니다\n",  
                    LIFE, cup);  
System.out.printf("평생(%d 년) Coffee 마시는데 %,d 원이 소비되었다",  
                    LIFE, money);  
}
```

# 급여 계산

- 다음은 홍길동의 급여이다. 다음 조건을 가지고 총 수입액, 세금, 실 임금액을 출력해보자
  - 본봉은 4,500,000 원
  - 보너스는 1,845,550 원
  - 총 수입액 = 본봉 + 보너스
  - 세금은 총 수입액의 8%
    - 세금 =
    - 세금은 10원 단위는 버림
  - 실 임금액 = 총 수입액 - 세금



# 급여 계산

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 본봉(bonBong) – int (4,500,000 원)

- 보너스(bonus) – int (1,845,550 원)

## ■ 출력

- 총 수입액(total) – int

- 세금(tax) – int

- 실 수입액(income) – int

## ■ 계산 방법

- 총 수입액 = 본봉 + 보너스

- 세금 = 총 수입액 \* (8.0 / 100)

(세금 계산 과정에서 double이 발생됨)

- 세금은 10원 단위는 버림 (세금 / 10 -> 세금 \* 10)

- 실 임금액 = 총 수입액 - 세금

매우 중요

# 급여 계산

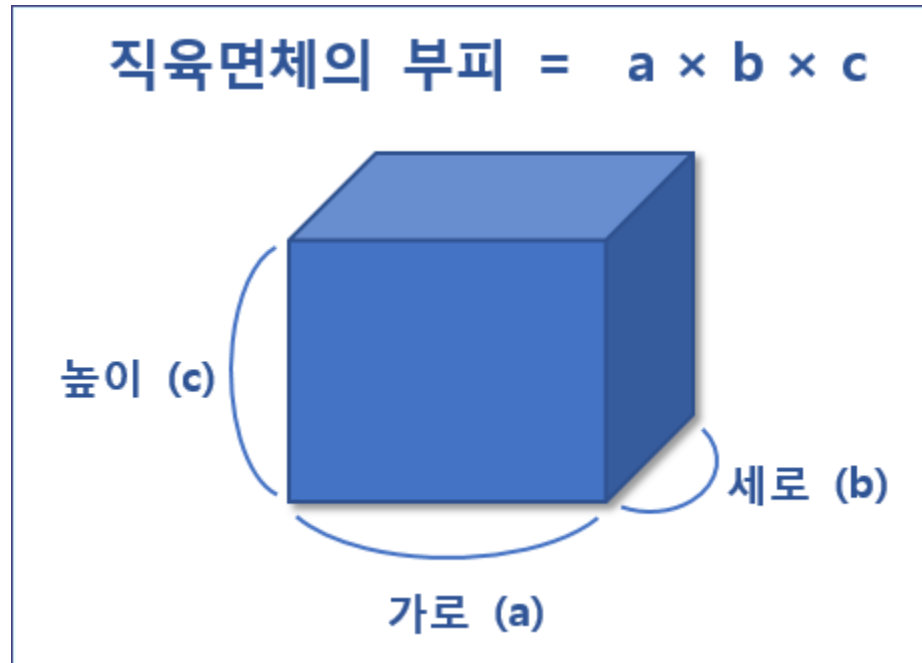
```
public static void main(String[] args) {  
    int total, tax, income;  
    int bonbong = 4500000;           //본봉은 450만원  
    int bonus = 1845550;             //보너스는 1845550원  
  
    total = bonbong + bonus;          //총 수입액은 본봉 + 보너스  
    tax = (int) (total * (8.0 / 100)); //세금은 총 수입액의 8%  
    tax /= 10;  
    tax *= 10;  
    income = total - tax;             //실 수입액은 총 수입액 - 세금  
  
    System.out.printf("홍길동의 총 수입액 : %,d 원\n", total);  
    System.out.printf("홍길동의 세금 : %,d 원\n", tax);  
    System.out.printf("홍길동의 실 수입액 : %,d 원\n", income);  
}
```

홍길동의 총 수입액 : 6,345,550 원  
홍길동의 세금 : 507,640 원  
홍길동의 실 수입액 : 5,837,910 원



# 육면체 부피

- 육면체의 부피를 계산하는 Program을 작성하여라.



# 육면체 부피

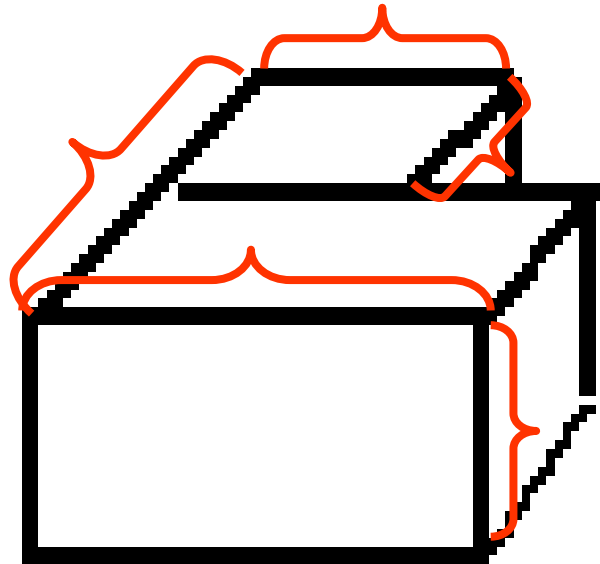
- 문제 분석
  - Data는 무엇이고, Information는 무엇인가?
- 입력
  - 육면체 가로 길이 - int
  - 육면체 세로 길이 - int
  - 육면체 높이 길이 - int
- 출력
  - 부피 - int
- 계산 방법
  - 공식 활용

# 육면체 부피

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int width, length, height;  
    int volume;  
  
    System.out.print("직육면체의 가로 세로 높이 : ");  
    width = keyboard.nextInt();  
    length = keyboard.nextInt();  
    height = keyboard.nextInt();  
  
    volume = width * length * height;  
  
    System.out.printf("\n 직육면체 가로 : %,10d\n", width);  
    System.out.printf("\n 직육면체 세로 : %,10d\n", length);  
    System.out.printf("\n 직육면체 높이 : %,10d\n", height);  
    System.out.printf("\n 직육면체 부피 : %,10d\n", volume);  
}
```

# 육면체 부피[심화]

- 다음과 같은 육면체의 표면적과, 부피를 구하시오. 어떻게 풀어야 할까? 문제 풀이 과정을 차근차근, 꼼꼼히 쓰시오.



# 육면체 부피[심화]

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 작은 육면체 가로, 세로, 높이 - int
- 큰 육면체 가로, 세로, 높이 - int
- 작은 육면체의 길이는 큰 육면체보다 작아야 함

## ■ 출력

- 표면적 = 큰 육면체 표면적 + 작은 육면체 표면적 - ?
- 부피 = 작은 육면체 부피 + 큰 육면체 부피

## ■ 계산 방법

- 표면적과 부피 구하는 공식

# 육면체 부피[심화]