

JAVA 프로그램 실습

형변환

경북대학교
소프트웨어융합과
배희호 교수

생각해 봅시다

- Computer에서 다음을 원칙적으로 계산할 수 있을까요?

$$5 + 5 = ?$$

$$5 + 5L = ?$$

$$5 + 5.5 = ?$$

$$5.5 + 5.5 = ?$$

생각해 봅시다

■ $\text{sum} = 3 + 4;$

1. 3을 2진수로 변환 \rightarrow 000...00000011

\rightarrow 임시 Memory(Register)에 저장

2. 4를 2진수로 변환 \rightarrow 000...00000100

\rightarrow 임시 Memory(Register)에 저장

3. 두 수를 더하여 임시 Memory(Register)에 저장

4. 임시 Memory(Register)의 내용을 sum으로 지정된 공간에 복사

생각해 봅시다

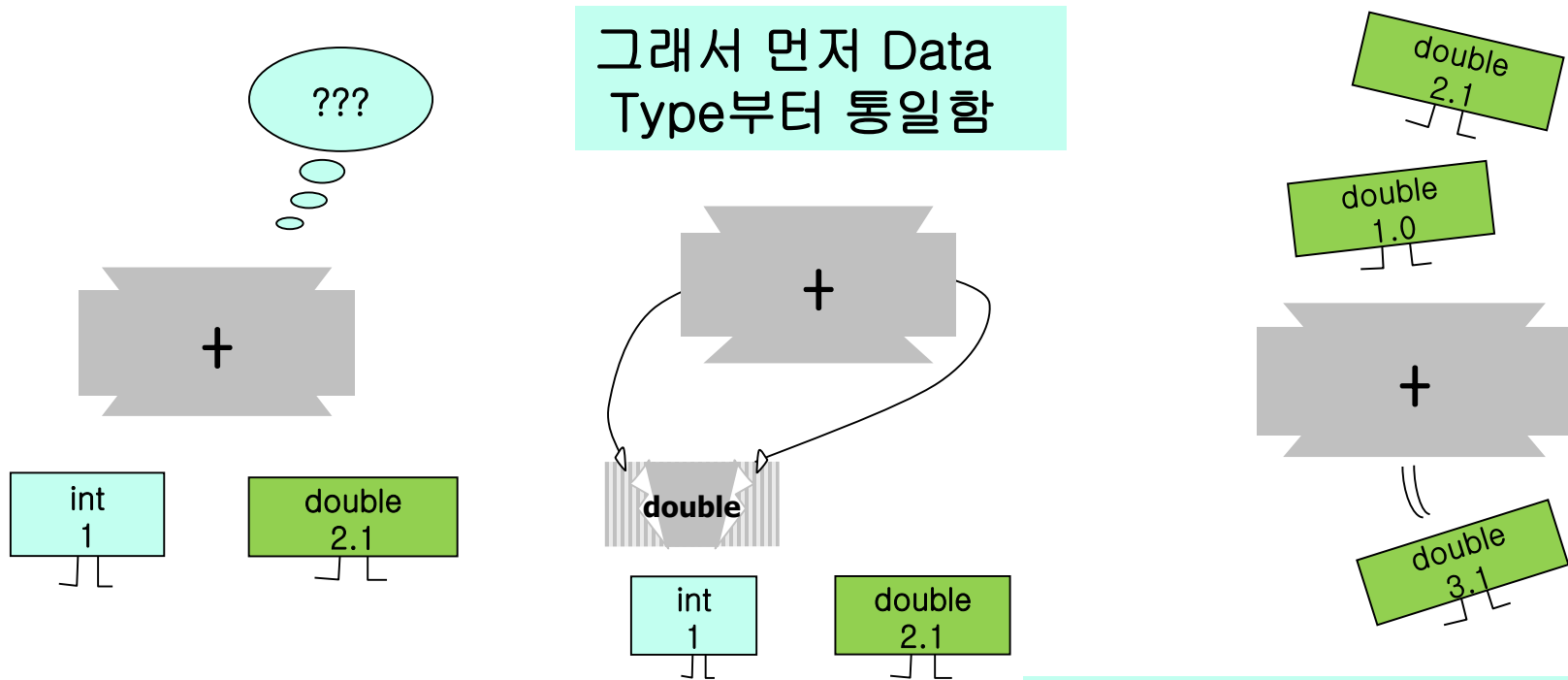
- 하나의 연산 수식에서 피 연산자의 Data Type이 서로 다를 경우 **원칙적으로 연산은 수행되지 않음**
- 일반적으로 연산 수식을 표현하기 위해 사용하는 변수나 상수들의 Type이 서로 같은 동일형보다는 서로 다른 혼합형(混血型) 수식이 많음
- JAVA 언어에서 혼합형의 수식을 사용하여도 되지만 그로 인하여 발생하는 오류의 책임은 **전적으로 사용자에게 있음**
- Primitive Data Type에 대해서는 관대하게도 Data Type을 **자동적으로 변환하여 처리함**

생각해 봅시다

- 숫자 자료형과 관련된 몇 가지 규칙
 - byte와 short 형의 값은 연산되기 이전에 int 형으로 변환되어 수행 (형 단순화)
 - 길이가 서로 다른 두 수의 연산을 수행할 경우
 - 길이가 작은 쪽을 큰 쪽으로 바꾸어 수행 (형 변환)
 - 길이를 초과하는 값을 저장하려고 할 경우
 - ArithmeticException이 발생하지 않는 대신 wrap으로 처리
- ArithmeticException
 - 어떤 수를 0으로 나누려고 하는 경우 발생

Data Types과 Operator

- Data Type과 밀접한 관련이 있는 연산자의 기능

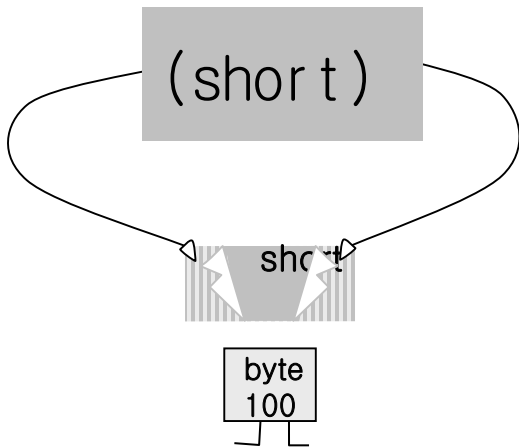


연산자는 Type이 다른 Data를 처리할 수 없음

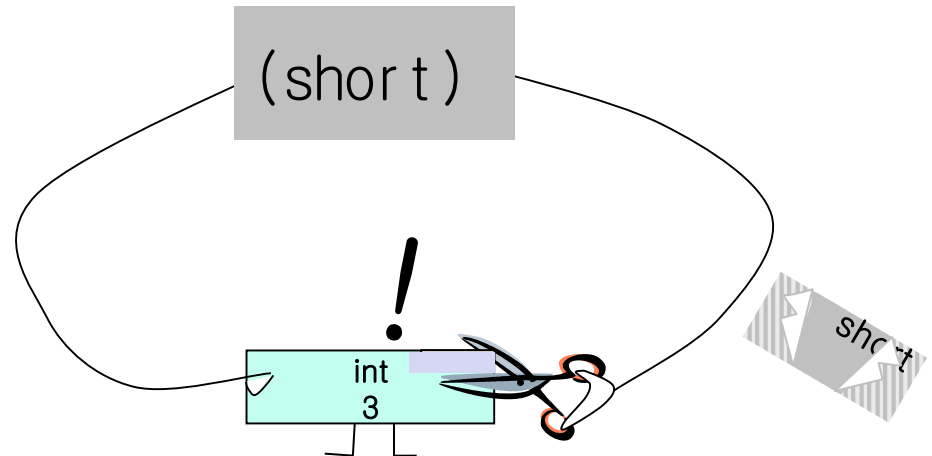
double 타입과 double 타입을 더하여 double 타입의 결과를 산출

Data Types과 Operator

■ Data Type을 바꾸는 Cast Operator



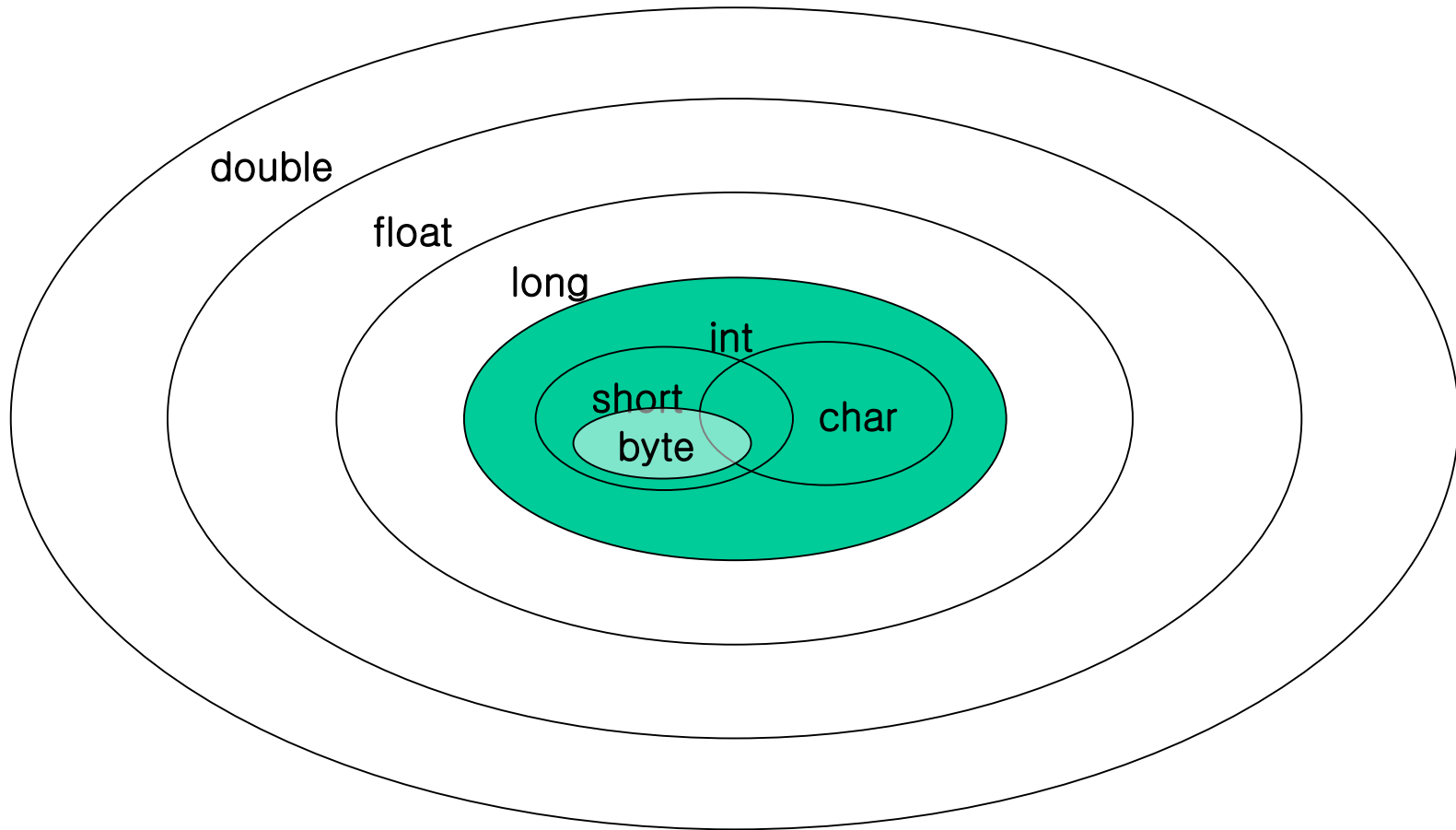
Cast Operator는 타입 변환을
전문으로 수행하는 연산자



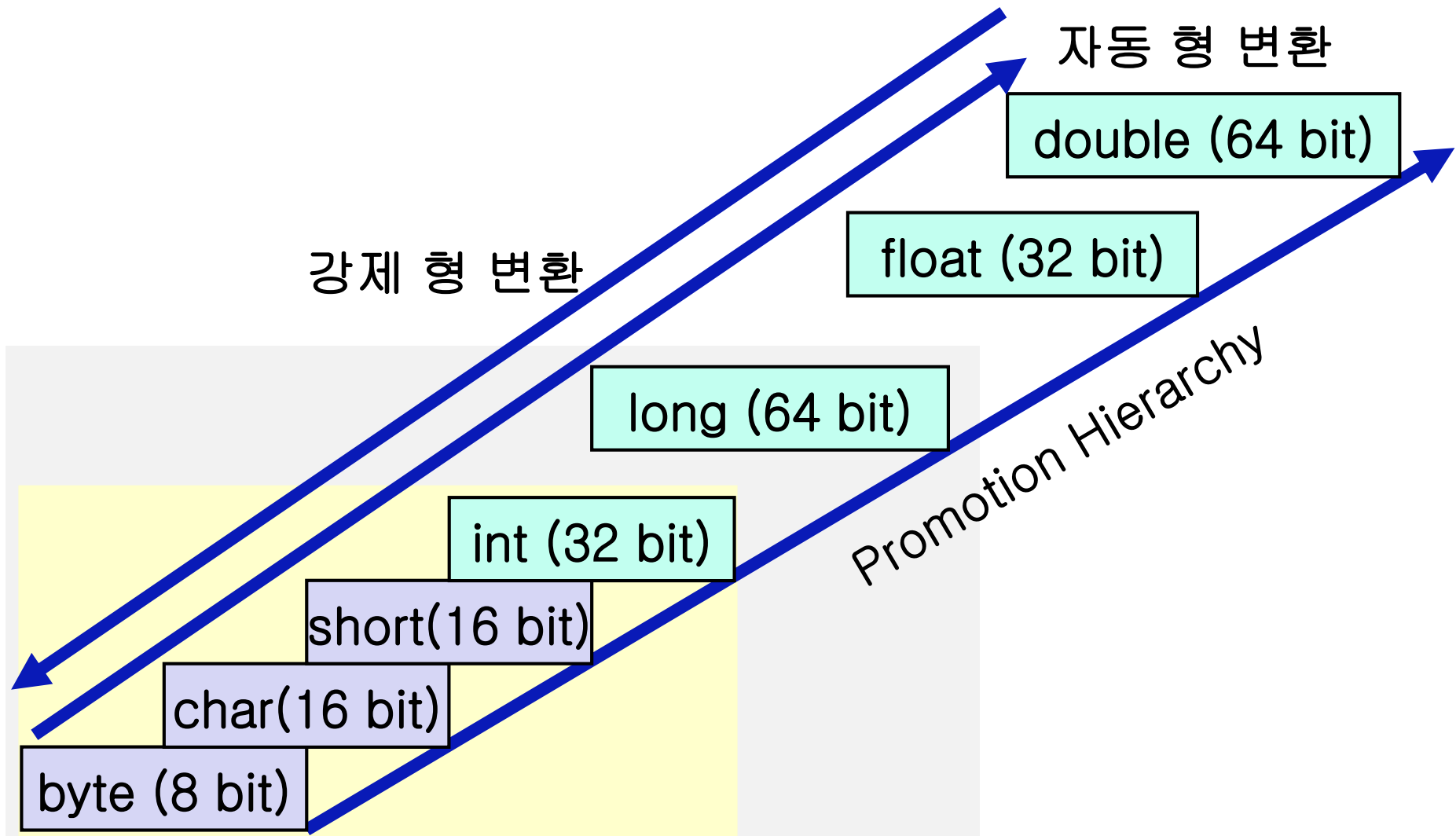
Cast Operator는 큰 범위 Type의
Data를
작은 범위 Type으로 바꿀 수도 있음

Data Types과 Operator

■ Data Type의 표현 범위



Data Type의 크기



Data Type의 크기

- 4 byte float 형이 더 큰 수를 표현할 수 있는 이유는 바로 부동 소수점 방식으로 표현하기 때문임
- 부동 소수점 방식에서는 지수부와 가수부를 나누고, 가수부에는 실제 값을 지수부에는 2 제곱을 얼마나 곱할지 표현하기 때문에 수의 표현 범위는 long 형보다 더 커지게 되는 것
- char 형이 byte 형보다 더 크니까 담을 수 있어 보이지만, 실제로 byte 형을 변환하여 char 형에 저장할 수는 없음
 - 왜냐하면 char 형은 음수를 표현할 수 없기 때문임
 - char 형은 Unicode와 ASCII Code를 표현할 목적을 가지고 있기 때문에 음수는 필요가 없어서 음수를 표현할 조합까지 양수를 더 많이 조합하는 것에 사용함
 - 하지만 byte 형은 비록 표현할 수 있는 Data 크기는 작아도 음수까지 표현하기 때문에, 음수를 표현할 수 없는 char 형에는 저장할 수 없는 것

Data Type 변환

- 이항 연산자는 연산을 수행하기 전에 피 연산자의 Type을 일치 시켜야 함

- int 형보다 크기가 작은 타입은 int로 형 단일화

byte, char, short → int

- 피연산자 중 표현 범위가 큰 Type으로 Type 변환

byte + byte → int + int → int

byte + short → int + int → int

char + char → int + int → int

int + byte → int + int → int

int + int → int

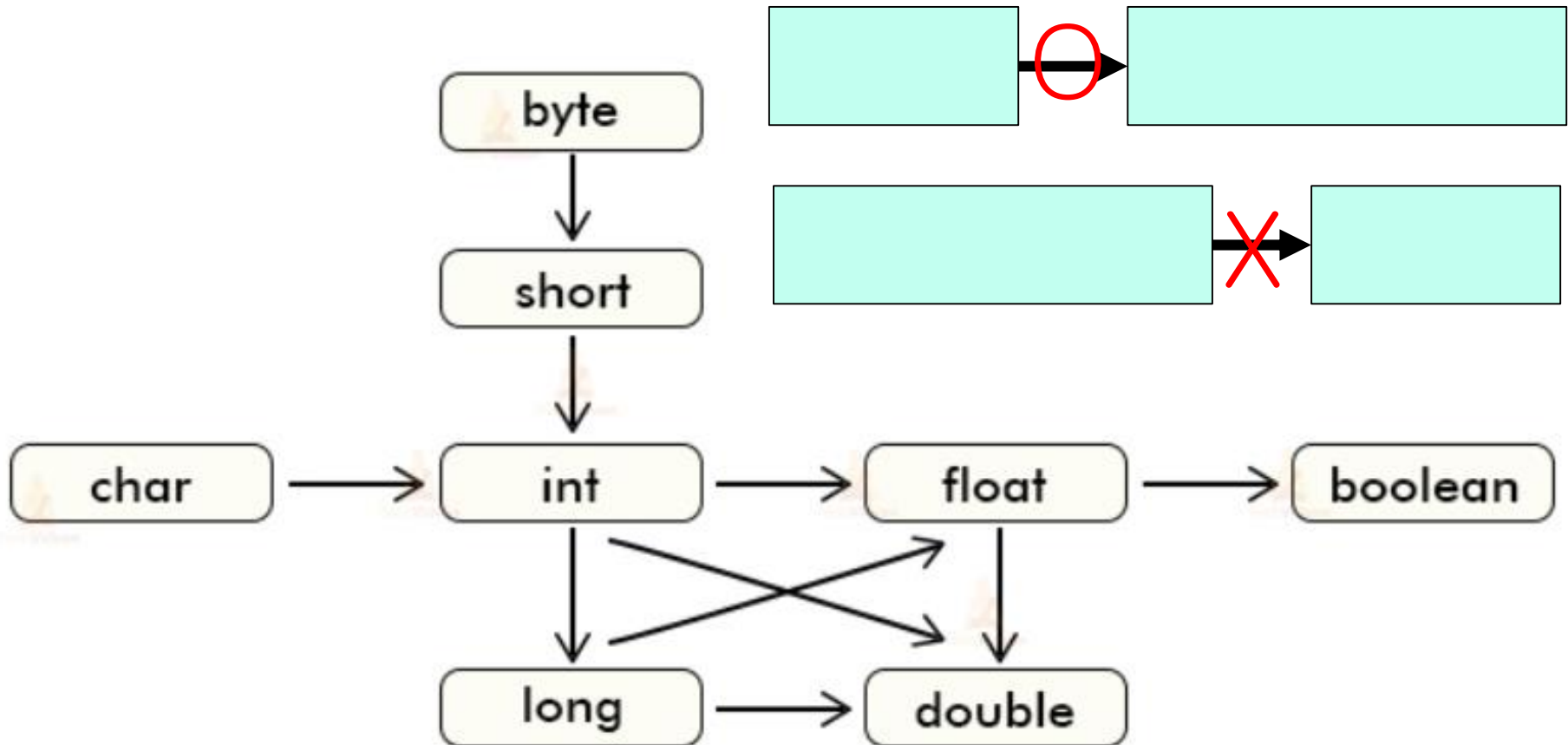
int + long → long + long = long

float + int → float + float → float

long + float → float + float → float

float + double → double + double → double

Data Type 변환



Data Type 변환

- Data Type 변환은 정보의 손실을 최대한 없애려는 방향으로 진행
- 모든 Data Type은 단항 상태에서 연산이 편리한 Type으로 변환
 - 이것은 System이 다양하게 세분된 Data Type을 처리하면 효율이 떨어지는 것을 방지 함
- Data Type 변환의 대상은 변수뿐만 아니라 상수도 동일하게 적용
- boolean을 제외한 나머지 7개의 Primitive Data Type들은 서로 Type 변환이 가능
- Primitive Data Type과 Reference Data Type 간에는 서로 Type 변환이 되지 않음
- 서로 다른 Type의 변수간의 연산에는 Type 변환이 요구되지만, 값의 범위가 작은 Type에서 큰 Type으로의 변환은 생략할 수 있음

Data Type 변환

- Data Type 변환 종류
 - Data Type의 크기 방향

확대 변환 (widening conversion)	✓ 혼합 수식에서의 변환 ✓ 작은 자료형의 값을 큰 자료형의 값으로 변환
축소 변환 (narrowing conversion)	✓ cast 연산자 사용 변환 ✓ 큰 자료형의 값을 작은 자료형의 값으로 변환 ✓ 정보손실 가능성

- 연산자가 서로 다른 Type의 피 연산자를 가질 때 형 변환이 필요
- 일반적으로 저장 공간의 크기가 작은 피 연산자를 큰 피 연산자 Type으로 변경

Data Type 변환

■ Data Type 변환 종류

■ Data Type 변환의 주체

자동 데이터형 변환	✓ 컴파일러에 의해 자동 수행 ✓ 혼합 수식에서의 변환
강제 데이터형 변환	✓ cast 연산자 사용 변환 ✓ 프로그래머가 형 변환을 명시

Data Type 변환

■ 강제 Data Type 변환

변환	수식	결과
int -> char	(char) 65	'A'
char -> int	(int) 'A'	65
float -> int	(int) 2.6f	2
int -> float	(float) 10	10f
double -> int	(int) 2.6	2
int -> double	(double) 10	10

Data Type 변환

■ int to String

```
String str = Integer.toString(i);  
String str = "" + i;
```

■ long to String

```
String str = Long.toString(l);
```

■ float to String

```
String str = Float.toString(f);
```

■ double to String

```
String str = Double.toString(d);
```

Data Type 변환

■ String to int

```
int i = Integer.parseInt(str);  
int i = Integer.valueOf(str).intValue();
```

■ String to long

```
long l = Long.parseLong(str);  
long l = Long.valueOf(str).longValue();
```

■ String to float

```
float f = Float.valueOf(str).floatValue();
```

■ String to double

```
double d = Double.valueOf(str).doubleValue();
```

Data Type 변환

- ASCII Code to String

```
String char = new Character((char)i).toString();
```

- Integer to ASCII Code

```
int i = (int) c;
```

- Integer to boolean

```
boolean b = (i != 0);
```

- boolean to Integer

```
int i = (b)? 1 : 0;
```

Data Type 변환

■ Decimal to Binary

```
String binstr = Integer.toString(i, 2);
```

■ Decimal to Hexadecimal

```
String hexstr = Integer.toString(i, 16);  
String hexstr = Integer.toHexString(i);  
Integer.toHexString(0x10000 | i).substring(1).toUpperCase();
```

■ Hexadecimal(String) to int

```
int i = Integer.parseInt(hexstr, 16);  
int i = Integer.parseInt(hexstr, 16);
```

■ String to double

```
double d = Double.parseDouble(str);
```

Data Type 변환

■ ASCII Code to String

```
String char = new Character((char)i).toString();
```

■ Integer to ASCII Code

```
int i = (int) c;
```

■ Integer to boolean

```
boolean b = (i != 0);
```

■ boolean to Integer

```
int i = (b)? 1 : 0;
```

Data Type 예제 1

- 연산자를 중심으로 양쪽 항의 Data Type이 같아야 함

```
int length;  
long height;  
float width;
```

```
length = 25;  
height = 10L;  
width = 17.5f;  
width = 17f;
```

연산자를 중심으로 양쪽 항의
데이터 형이 같아야 함

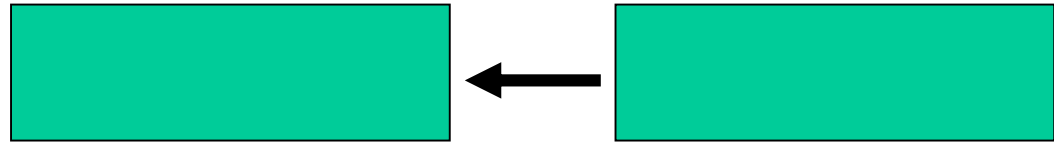
데이터 형이 다르면 하나의
형으로 통일 한다

```
width = (float) height + length;
```

Data Type 예제 1

- 대입 연산자의 양 항의 Data Type은 일치해야 함

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

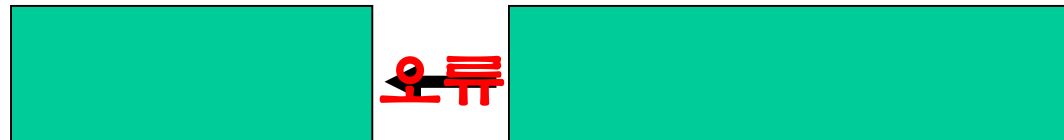


- 대입 연산자의 양 항의 Data Type이 일치하지 않는 경우

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



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



Data Type 예제 2

- 다음 Program의 Error의 원인을 설명하고 수정하여라

```
public static void main(String[] args) {  
    int test1 = 128;  
    byte test2 = 128; (1)  
    byte test3 = 127;  
    short test4 = 32768; (2)  
    short test5 = 32767;  
    long test6 = 32767;  
    char test7 = -65; (3)  
    char test8 = 65;  
    char test9 = 'A';  
    double test10 = 1.111;  
    float test11 = 1.111; (4)  
}
```


Data Type 예제 2

- byte와 int는 모두 정수 형으로 각각 1 byte(8 bit)와 4 byte(32 bit)의 크기를 가지며, 표현할 수 있는 값의 범위는 byte가 $-2^7 \sim 2^7 - 1$ ($-128 \sim 127$), int가 $-2^{31} \sim 2^{31} - 1$
- short과 char은 모두 2 byte의 크기를 갖지만, short의 범위는 $-2^{15} \sim 2^{15} - 1$ ($-32,768 \sim 32,767$)이고 char의 범위는 $0 \sim 2^{16} - 1$ ($0 \sim 65,535$)이므로 서로 범위가 달라서 둘 중 어느 쪽으로의 형 변환도 값 손실이 발생할 수 있으므로 자동적으로 형 변환이 수행될 수 없음
- char형은 10진수로 0~65,535의 코드 값을 가짐
- 문자 'A'의 코드는 10진수로 65

Data Type 예제 2

- 실수는 float 또는 double을 Data Type으로 넣을 수 있는데 그 값은 기본적으로 double
 - 정수는 기본이 int지만 실수에서의 기본은 double이라는 뜻
- 즉, double의 크기를 가진 1.111은 float가 될 수 없음
 - float f의 float을 double로 변경
 - 1.111 뒤에 f를 붙여 1.111f 라고 입력하면 입력한 실수를 float으로 변환

Data Type 예제 3

- 다음 Program의 Error의 원인을 설명하고 수정하여라

```
public static void main(String[] args) {  
    int test1 = 2;  
    int test2 = 3;  
    int test3 = test1 + test2;          (1)  
  
    short test4 = 2;  
    short test5 = 3;  
    short test6 = test4 + test5;        (2)  
  
    byte test7 = 2;  
    byte test8 = 3;  
    byte test9 = test7 + test8;         (3)  
}
```

Data Type 예제 3

- (2)에서 Error가 발생

- test4 + test5에서 각 항이 short 형이고 같으므로 이상이 없는 것으로 생각하지만 형 단일화를 먼저하므로 short형은 int로 형 변환하고, + 연산 결과는 int형이 되므로 오류가 발생

- (3)에서 Error가 발생

- test7 + test8에서 각 항이 byte 형이고 같으므로 이상이 없는 것으로 생각하지만 형 단일화를 먼저하므로 byte 형은 int로 형 변환하고, + 연산 결과는 int형이 되므로 오류가 발생

Data Type 예제 4

- 다음 Program의 Error의 원인을 설명하고 수정하여라

```
1. public static void main(String[] args) {  
2.     int test1 = 3;  
3.     short test2 = 3;  
4.     boolean test3 = true;  
5.     char test4 = 'A';  
6.  
7.     int test5 = test1 + test3;      (1)  
8.     int test6 = test2 + test4;      (2)  
9. }
```

Data Type 예제 4

- (1)에서 오류가 발생
 - test1 + test3에서 각 항이 int형과 boolean형 임
 - boolean 형은 형 변환을 할 수 없음
 - C 언어와 같이 'true'는 1, 'false'를 0으로 변환하여 연산하려면 `int test5 = test1 + (test3 ? 1 : 0);`

Data Type 예제 5

- 다음 Program의 Error의 원인을 설명하고 수정하여라

```
public static void main(String[] args) {  
    int result1;  
    int result2;  
    float result3;  
    double result4;  
  
    result1 = 12 + 34.5;           (1)  
    result2 = 12 + 34L;           (2)  
    result3 = 12 + 34.5;          (3)  
    result4 = 12 + 34.5f;  
}
```

Data Type 예제 5

- (1)에서 오류가 발생
 - $12 + 34.5$ 에서 int 형 + double 형이므로 int 형인 12를 double 형으로 변환하여 12.0으로 변환하여 형 단일화하여 결과를 double 형인데 int 형에 대입하므로 Error
- (2)에서 오류
- (3)에서 오류

Data Type 예제 6

- 아래 Program의 Compile Error를 수정하고, 설명하여라

```
public class Question1 {  
    public static void main(String[] args) {  
        int a = 500000;  
        float f = 50.6; (1)  
        double d = f;  
        double b = a;  
        short c = a; (2)  
        long l = a;  
        int test = d; (3)  
        System.out.println("test = " + test);  
        System.out.println("d = " + d);  
    }  
}
```

Data Type 예제 6

- (1)번 오류
 - 50.6은 double 형 Literal임
 - 이것을 보다 작은 float 기억장소에 저장할 수 없음
 - Cast 연산자가 필요
- (2)번 오류
- (3)번 오류

자동 Data Type 변환

- 연산 시 자동 타입 변환 정리
 - 정수와 실수 연산 = 실수형 타입 (double)
 - 정수와 정수 연산 = int 타입
 - 실수와 실수 연산 = double 타입
 - 큰 타입과 작은 타입 연산 = 큰 타입

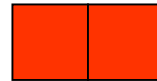
자동 Data Type 변환

float

int

float

total = count * weight;



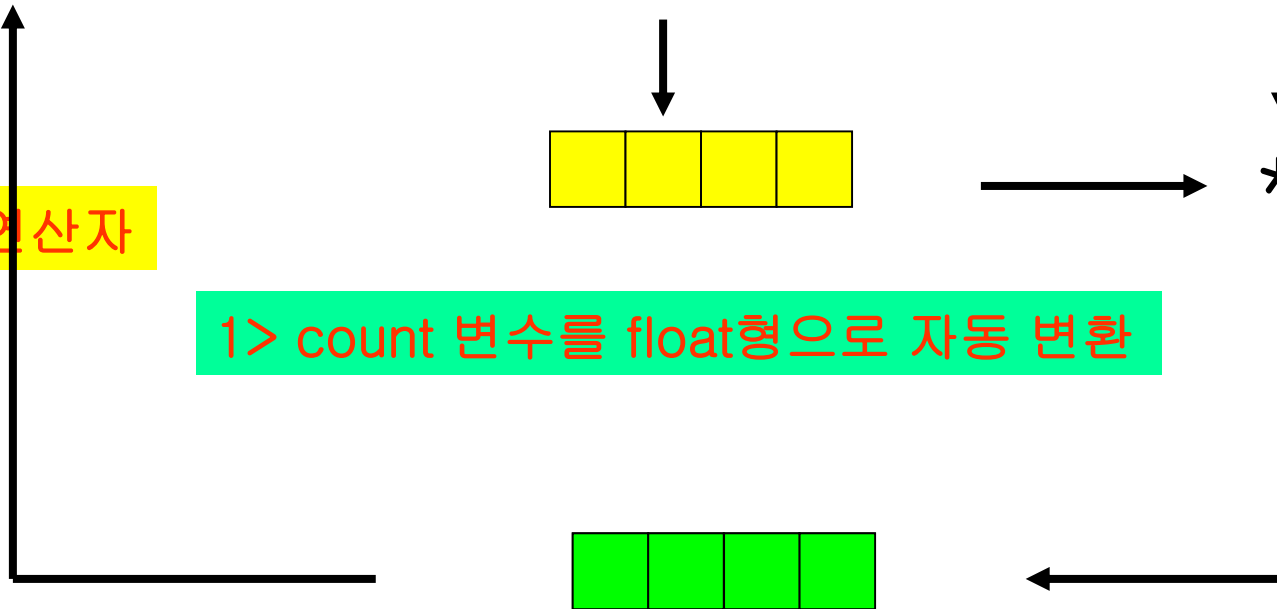
*

4> 대입 연산자

1> count 변수를 float형으로 자동 변환

2> 연산

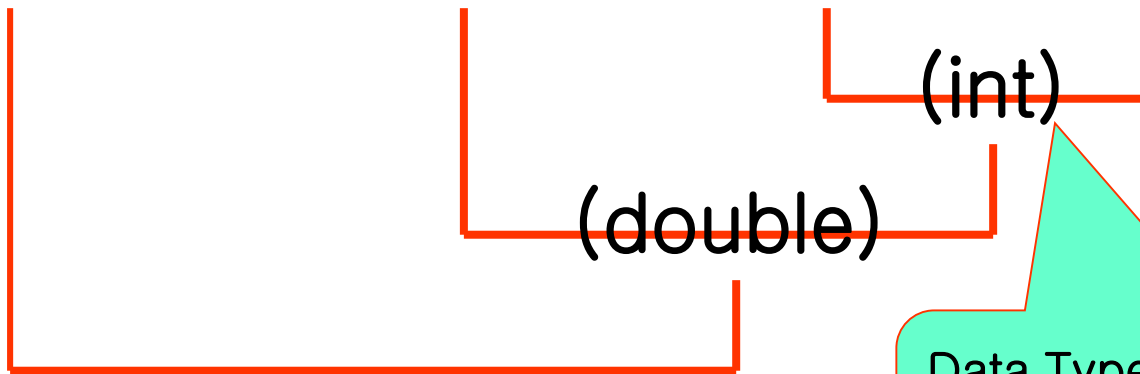
3> 연산 결과를 float형의 임시 변수에 기억



자동 Data Type 변환

- Data Type을 같도록 한 후에 연산 수행

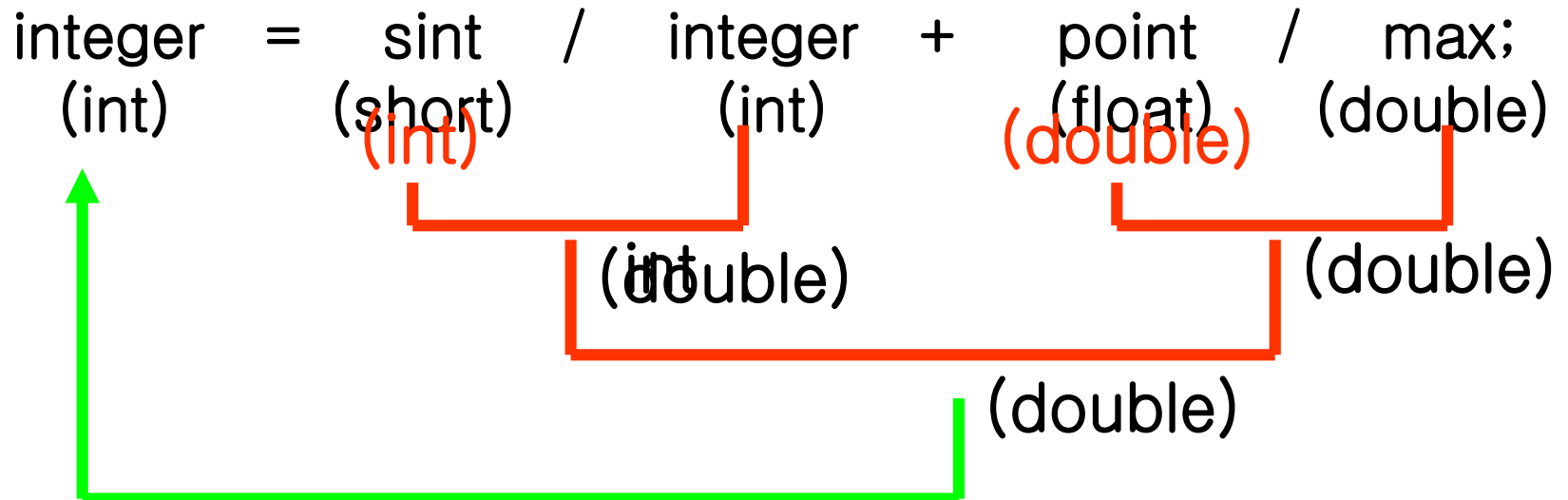
```
test      =  1.0  +  2  *  3
(float)    (double) (int) (int)
```



Data Type이 일치하지 않으므로
일시적으로 int 형을 보다 우선
순위가 높은 double 형으로 변환

자동 Data Type 변환

```
short int sint;  
int integer;  
float point;  
double max;
```



자동 Data Type 변환

- 대입문(치환문)에서 연산의 최종 결과는 저장할 변수의 Data Type으로 변환

1) float, double 형 변수에 char, int형 상수를 대입

```
float apple = 35;
```



```
apple = 35.0
```



2) char, int 형 변수에 float, double형 상수를 대입

```
int apple = (int) 35.89;
```



```
apple = 35
```



손실 발생

자동 Data Type 변환

3) 동일 Data Type에서 크기가 작은 형 변수에 큰 형의 상수를 대입하는 경우

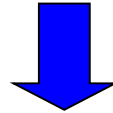
```
short apple = (short) 40000L;
```



```
apple = -25536
```

0000 0000 0000 0000 1001 1100 0100 0000

40,000



1001 1100 0100 0000

-25,536

하위 비트부터 Data가 전달되면서
상위 비트의 손실이 발생



자동 Data Type 변환

4) 동일 Data Type에서 크기가 큰 형 변수에 작은형의 상수를 대입하는 경우

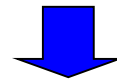
(1) 부호 비트가 양수(0)이고, 부호비트 확장의 경우

```
long int apple = 30;
```



apple = 30

0001 1100 0100 0000 30



0000 0000 0000 0000 0001 1100 0100 0000 30

상위 비트를 모두 0으로 확장하면 데이터 손실 없음

자동 Data Type 변환

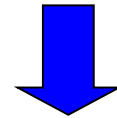
(2) 부호 비트가 음수(1)이고, 부호 비트 확장의 경우

```
long int apple = -192;
```



```
apple = -192
```

1111 1111 0100 0000 -192



1111 1111 1111 1111 1111 1111 0100 0000 -192

상위 비트를 모두 1로 확장하면 데이터 손실 없음

자동 Data Type 변환 예제 1

선언

```
char c;    short s;    int i;  
long l;    unsigned u;  unsigned long ul;  
float f;   double d;   long double ld;
```

수식	형	수식	형
c - s / i	int	u * 7 - i	
u * 2.0 - i		f * 7 - i	
c + 3		7 * s * ul	
c + 5.0		ld + c	
d + s		u - ul	
2 * i / l		u - l	

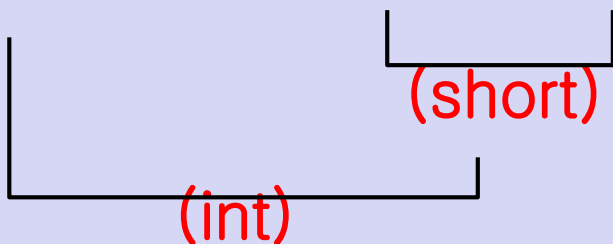


자동 Data Type 변환 예제 2

- 다음 연수 수식을 평가하여라.(프로모션)

```
char c = 'A';  
short s = 1; int i = 2;
```

```
  i    =    ( c    +    s );           // i = 66  
(int)      (char) (short)
```



자동 Data Type 변환 예제 3

■ 다음 연수 수식을 평가하여라.(프로모션)

```
short s = 1;  
int i = 2; long l = 10;
```

```
l      =      s      +      i      ;           // l = 3  
(long)   (short)  (int)  
          └──┬──┘  
              (int)  
          └──────────┘  
              (long)
```

자동 Data Type 변환 예제 4

■ 다음 연수 수식을 평가하여라.(프로모션)

```
float f = 2.1f; double d = 3.2;
```

```
d = f + d; // d = 5.3
```

(double) (float) (double)

(double)

자동 Data Type 변환 예제 5

- 다음 Program의 Compile Error를 수정하고, 설명하여라

```
public static void main(String[] args) {  
    byte num1 = 2, num2 = 3, sum;  
  
    sum = num1 + num2; (1)  
  
    System.out.printf("%d + %d = %d\n", num1, num2, sum);  
}
```

자동 Data Type 변환 예제 5

- (1)에서 오류
 - 자동 형변환 중 형 단일화 하므로 오류

자동 Data Type 변환 예제 6

- 다음 Program의 Compile Error를 수정하고, 설명하여라

```
public class Question1 {  
    public static void main(String[] args) {  
        short s1 = 1, s2 = 2;  
        int i1 = 3, i2 = 4;  
  
        short result = s1 + s2;          (1)  
        int result1 = i1 + i2;  
        System.out.println("result = " + result);  
        System.out.println("result1 = " + result1);  
    }  
}
```

자동 Data Type 변환 예제 6

- (1)에서 오류
 - 2 byte 의 s1과 s2의 합을 구하는 문제인데 그 과정에서 $s1 + s2$ 는 단항 상태에서 형 단일화를 위해서 자동형 변환 되어 int로 각각 변환
 - JAVA 환경에서는 int가 가장 빠르게 작동하기 때문인데, 결국 int로 변경된 $s1 + s2$ 는 결과를 int 임
 - int를 result(short 형)에 대입할 수 없음
 - 오류를 수정하는 방법은 2가지
 - short result를 int result로 바꾸면 됨
 - (short) ($s1 + s2$)로 연산 앞에 (short)를 입력해서 int로의 변환을 제한하는 방법. 여기서 소괄호로 표현한 내용을 cast 연산자 라고 함

자동 Data Type 변환 예제 7

- 다음 Program의 Compile Error를 수정하여라.

```
public class Question4{  
    public static void main(String[] args) {  
        int i = 1;  
        short s = 2;  
        boolean b = true;  
        char c = 'C';  
        int result = i + b; (1)  
        int result1 = s + c;  
        System.out.println("result = " + result);  
        System.out.println("result1 = " + result1);  
    }  
}
```

자동 Data Type 변환 예제 7

- (1)에서 오류
 - boolean은 1 Byte의 논리값을 저장
 - 논리값은 숫자와 연산 될 수 없기 때문에 오류가 남
- char는 숫자로 표현할 수 있음
- 보통 연산에 사용하지는 않고 ASCII Code나 Unicode에 활용하지만 숫자로 표현은 가능
- char는 2 Byte를 가지기에 short s의 2 Byte와 연산해서 int로 자동 형 변환 하더라도 int result와 정상 작동

자동 Data Type 변환 예제 8

- 다음 Program의 Compile Error를 수정하여라.

```
public class Question2 {  
    public static void main(String[] args) {  
        float result;  
        result = 12 + 34.5;          (1)  
  
        System.out.println("12 + 34.5 = " + (12 + 34.5));  
        System.out.println("12 + 34.5 = " + result);  
    }  
}
```

자동 Data Type 변환 예제 8

■ (1)의 오류

- $12 + 34.5$ 의 연산은 `int + double`로 `int`를 `double`로 변환하여 `double + double = double`이므로 `float`에 대입할 수 없음
- 형 변환 종류는 다른 자료형 간의 연산 시 어떤 자료형 그릇 크기에 맞추느냐에 따라 사용처가 달라지게 됨
 - 작은 자료 형으로 맞출 시
 - 작은 자료 형으로 강제 형 변환
 - 큰 자료 형으로 맞출 시
 - 큰 자료 형으로 자동 형 변환

강제 Data Type 변환

- Type 변환 연산자 (Cast 연산자)
 - 강제로 Type 변환을 하고자 할 경우에는 Type 변환 연산자를 사용
 - 상수나 변수 앞에 바꾸고자 하는 Type 변환 연산자를 () 안에 기술하면 됨

연산자	설명
(기본 데이터형)	기본 데이터 형을 사용한 문장에서 일시적으로 형 변환이 발생(기본 데이터형: double, float, long, int, short, char, byte, boolean)

■ Type 변환 연산자의 예

(char) 65	문자형 A
(int) 2.5	2
(float) 4 * 5	20.0
21 / (int) 3.4	7

강제 Data Type 변환

- 더 큰 크기의 변수로 값을 이동하는 변환 (확대 변환)

```
double value = (double) 100;
```

정수 100이 변수 value에
100.0으로 형 변환되어서 저장

강제 Data Type 변환

■ 형 변환이 필요한 이유?

```
double myDouble = 33.3;
```

```
int result;
```

```
...
```

```
result = (int) (myDouble * 3);
```

99.9

소숫점 이하 0.9 손실

강제 Data Type 변환 예제 1

```
char c = 'A';  
float f = 2.1f;  
double d = 3.2;
```

```
f = (float) ( f + d );    // f = 5.3
```

(float) (float) (double)

(double)

(float)

강제 Data Type 변환 예제 2

■ 다음 Program의 실행 결과를 설명하여라.

```
public static void main(String[] args) {
```

```
    byte b;
```

```
    int i = 259;
```

```
    double d = 525.428;
```

```
    b = (byte) i;
```

```
    i = (int) d;
```

```
    System.out.println("축소 형 변환 결과");
```

```
    System.out.println("int 259를 byte로 : " + b);
```

```
    System.out.println("double 525.428을 int로 : " + i);
```

```
    b = (byte) d;
```

```
    System.out.println("double 525.428을 byte로 : " + b);
```

```
}
```

축소 형 변환 결과

int 259를 byte로 : 3

double 525.428을 int로 : 525

double 525.428을 byte로 : 13



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

강제 Data Type 변환 예제 2

- int -> byte로 변환
 - JAVA에서 int와 byte는 모두 기본 유형이며 숫자 값을 저장하는 데 사용
 - 둘 다 부호 있는 값과 부호 없는 값을 저장하는 데 사용되지만 저장 범위는 다름
 - byte 범위는 -128 ~ 127(256)이고, int 범위는 -2,147,483,648 ~ 2,147,483,647
 - 256으로 나누고 나머지를 구하면 됨
- double -> int로 변환
 - 정수 부분만 int로 변환

강제 Data Type 변환 예제 3

■ 다음 Program의 실행 결과를 설명하여라.

```
public static void main(String[] args) {  
    double d = 100.0;  
    int i = 100;  
  
    int result =(int) (i + d);  
    int test = (int) (3.141592f * 1000) / 1000;  
  
    System.out.println("d = " + d);  
    System.out.println("i = " + i);  
    System.out.println("result = " + result);  
    System.out.println("test = " + test);  
}
```

```
d = 100.0  
i = 100  
result = 200  
test = 3
```

강제 Data Type 변환 예제 3

- 연산과정에서 자동 형 변환이 발생하여도 변수의 값이 변경되는 것은 아님

강제 Data Type 변환 예제 4

```
public static void main(String[] args) {  
    int i;  
    double f;  
  
    f = (double) 5 / 4;  
    System.out.println(f);  
  
    f = (double) 5 / (double) 4;  
    System.out.println(f);  
  
    f = 5 / 4;  
    System.out.println(f);  
  
    i = (int) 1.3 + (int) 1.8;  
    System.out.println(i);  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

자동 Data Type 변환 예제 4

- 연산과정에서 자동 형 변환이 발생하여도 변수의 값이 변경되는 것은 아님

강제 Data Type 변환 예제 5

■ 다음 Program의 형 변환을 설명하여라

```
public static void main(String[] args) {
```

```
    byte b = 127;
```

```
    int i = 100;
```

```
    System.out.println(b+i);
```

```
    System.out.println(10/4);
```

```
    System.out.println(10.0/4);
```

```
    System.out.println((char) 0x12340041);
```

```
    System.out.println((byte) (b+i));
```

```
    System.out.println((int)2.9 + 1.8);
```

```
    System.out.println((int)(2.9 + 1.8));
```

```
    System.out.println((int)2.9 + (int)1.8);
```

```
}
```

227
2
2.5
A
-29
3.8
4
3



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

강제 Data Type 변환 예제 5

- 연산과정에서 자동 형 변환이 발생하여도 변수의 값이 변경되는 것은 아님

강제 Data Type 변환 예제 6

■ 다음 Program의 형 변환을 설명하여라

```
public static void main( String[] args ) {  
    int i;  
    byte b;  
    short s, temp=128;  
  
    b = (byte) temp; // narrowing conversion  
    System.out.println(b);  
  
    s = (short) (temp*100); // overflow 발생 가능성이  
                           // 있으므로 형변환 명시  
  
    System.out.println(s);  
    i= temp;  
    System.out.println(i);  
}
```

-128
12800
128



강제 Data Type 변환 예제 7

■ 다음 Program의 형 변환을 설명하여라

```
public static void main(String[] args){  
    byte b1 = 10;  
    byte b2 = 14;  
  
    byte result = (byte) (b1 + b2);  
    System.out.println(resultB);  
}
```

정수형 데이터 유형의 기본 유형은 int이기 때문에 byte형끼리 덧셈을 수행하더라도 그 결과값은 int가 된다. 따라서, 위 예제에서는 형변환을 하여 byte형 변수에 결과값을 할당하였다

강제 Data Type 변환 예제 8

■ 다음 Program의 형 변환을 설명하여라

```
public static void main(String[] args) {  
    char na = '나';  
    byte bch = (byte) na;  
    short sch = (short) na;  
  
    System.out.print("na = " + na );  
    System.out.print(", (byte)na = " + bch );  
    System.out.println(", (short)na = " + sch );  
    System.out.println("(int)na = " + (int) na );  
    System.out.println("(long)na = " + (long) na );  
    System.out.print("(char)45208 = " + (char) 45208 );  
    System.out.println(", (char)45209 = " + (char)(45208 + 1) );  
}
```

na = 나, (byte)na = -104, (short)na = -20328
(int)na = 45208
(long)na = 45208
(char)45208 = 나, (char)45209 = 낙

강제 Data Type 변환 예제 8

- 문자 자료형 na에 한글 문자 '나'를 저장
- 이 문자 변수 na는 명시적 변환을 사용하지 않고는 byte나 short 형으로 변환이 불가능
- 변수 na를 명시적으로 변환을 하면 byte나 short 형의 변수에 저장할 수 있으나, 오버플로우가 발생하여 정확한 유니코드 값을 얻을 수는 없음

강제 Data Type 변환 예제 9

■ 다음 Program의 결과를 설명하여라

```
public static void main(String[] args) {  
    int big = 1234567890;  
    float approx;  
  
    approx = (float) big;  
  
    System.out.println("approx = " + approx);  
    System.out.println("(int) approx = " + (int) approx);  
    System.out.println("difference = " + (big - (int) approx));  
}
```

```
approx = 1.234568E9  
(int) approx = 1234567936  
difference = -46
```

Data 값 상수

- JAVA에서 Data 값 검사를 위해 해당 Type의 최대값, 최소값 상수를 제공

기본 타입	최대값 상수	최소값 상수
byte	Byte.MAX_VALUE	Byte.MIN_VALUE
short	Short.MAX_VALUE	Short.MIN_VALUE
int	Int.MAX_VALUE	Int.MIN_VALUE
long	Long.MAX_VALUE	Long.MIN_VALUE
float	Float.MAX_VALUE	Float.MIN_VALUE
double	Double.MAX_VALUE	Double.MIN_VALUE

Data 값 상수 예제

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int num;  
    byte test;  
  
    System.out.print("하나의 정수 입력 : ");  
    num = keyboard.nextInt();  
  
    if ((num < Byte.MIN_VALUE) || (num > Byte.MAX_VALUE) ) {  
        System.out.println("byte 타입으로 변환 할 수 없음");  
    } else {  
        test = (byte) num;  
        System.out.println(num + "의 byte 형 변환 : " + test);  
    }  
}
```

도입 예제

- 나의 IQ는 128이고, 연봉은 32,767,000원 입니다. 이것을 다음과 같이 Program하여 그 결과를 설명하고, 올바르게 수정해보자.

```
public static void main(String[] args) {  
    byte IQ = (byte) 128;  
    System.out.println("나의 IQ는 " + IQ + " 입니다");  
    short salary = 32767;  
    System.out.printf("나의 연봉은 %,d 천원 입니다\n", salary);  
    byte increse = 100;  
    System.out.println("나의 연봉이 " + increse + " 천원 올랐습니다");  
    salary += increse;  
    System.out.printf("나의 연봉은 %,d 천원 입니다\n", salary);  
}
```

나의 IQ는 -128 입니다
나의 연봉은 32,767 천원 입니다
나의 연봉이 100 천원 올랐습니다
나의 연봉은 -32,669 천원 입니다

도입 예제

- 강제적 Type 변환에서의 정보의 손실
 - IQ의 값은 1280이 아닌 -128 출력

```
short salary = 32767;  
byte increment = 100;  
salary = (short) (salary + increment);
```

- 연산 수식 salary + increment에서 모두 int로 변환하여 더하기를 수행하므로 결과는 int 형 32,867이나, 이 값을 short 형으로 변환하여 저장하므로 2 byte만을 저장
- 그러므로 변수 salary에는 -32,669 값이 저장
- 만약 salary = (short) (salary + increment);를 salary += increment; 로 작성하여도 됨. 그러나 salary = salary + increment;는 오류가 발생 함

우리나라 대한민국

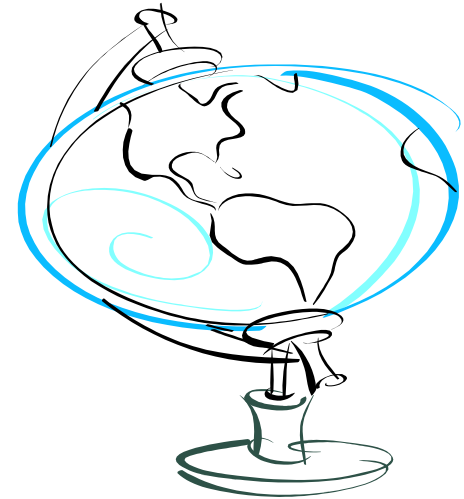
■ 대한민국을 기준으로 각 나라의 비율을 구하여라.

■ 대한민국의 면적 : $99,720\text{km}^2$

■ 북한의 면적 : $122,762\text{km}^2$

■ 미국의 면적 : $9,631,418\text{ km}^2$

■ 일본의 면적 : $377,873\text{km}^2$



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

우리나라 대한민국

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 각 국가의 면적 - long

- 출력

- 대한민국의 몇 배의 국토를 갖고 있는지 비율 - float

- 알고리즘

- 나누기 연산 (float)

우리나라 대한민국

```
public static void main(String[] args) {  
    final int SouthKOREA = 99720;  
    final int NorthKOREA = 122762;  
    final int USA = 9631418;  
    final int JAPAN = 377873;  
  
    System.out.printf("북한은 대한민국의 %.2f 배의 국토를 갖고 있음\n",  
                      (float) NorthKOREA / SouthKOREA);  
    System.out.printf("미국은 대한민국의 %.2f 배의 국토를 갖고 있음\n",  
                      (float) USA / SouthKOREA);  
    System.out.printf("일본은 대한민국의 %.2f 배의 국토를 갖고 있음\n",  
                      (float) JAPAN / SouthKOREA);  
}
```

1년의 시간

- 1년은 365.2422일이다. 이것은 며칠, 몇 시, 몇 분, 몇 초인가를 구하는 Program을 작성하여라.



1년의 시간

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

■ 입력

- 1년(year) – float (365.2422 일)

■ 출력

- 며칠, 몇 시간, 몇 분, 몇 초인지 출력

■ 계산 방법

- 하루(day) : 24 시간

- 시간(hour) : 60 분

- 분(minute) : 60 초

- 나머지 연산자와 나눗셈 연산자 이용

1년의 시간

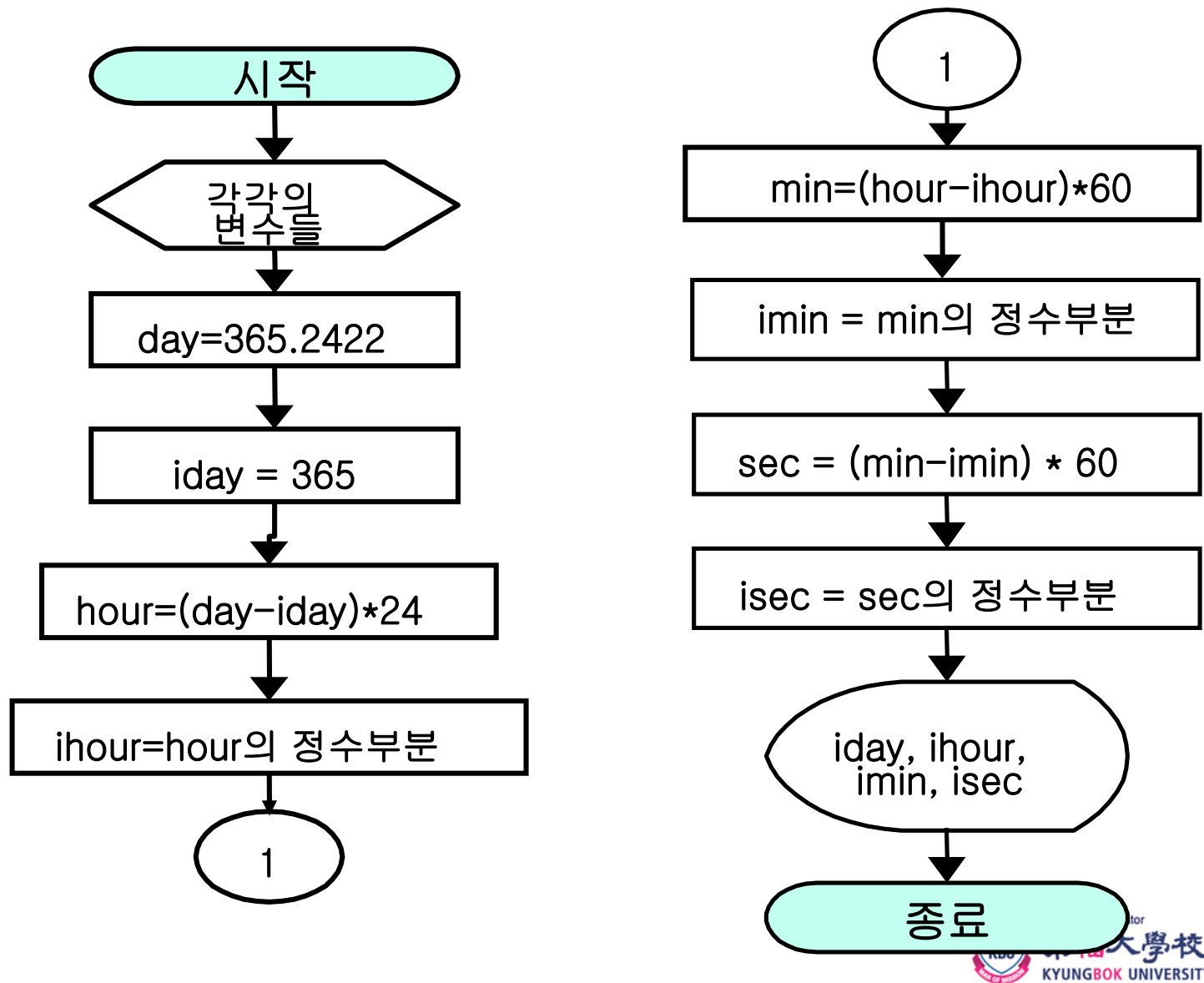
■ 문제 분석

1. 365.2422일에서 정수부분은 날짜가 됨
2. 나머지 0.2422부분은 시간으로 환산
3. 1일이 24시간이므로 0.2422일은 $24 \times .2422 = 5.8128$
4. 3단계에서 시간으로 환산하면 실수부분과 정수부분이 나온다(5.8128시간). 이를 정수와 실수로 분리하면 5시간 0.8128시간
5. 4단계에서 남은 실수부분의 시간은 다시 분으로 환산
6. 1시간은 60분 이므로 0.8128시간은 $0.8128 \times 60 = 48.768$ 분. 구하고 자 하는 분은 48분
7. 6단계에서 나머지 0.768분은 다시 초단위로 환산.
8. 1분 역시 60초이므로 0.768분은 $0.768 \times 60 = 46.08$ 초

1년의 시간

- 저장 변수
 - 주어진 날짜-day
 - 24시간으로 정확히 떨어지는 날짜 : iday
 - 시간 : hour
 - 60분으로 정확히 떨어지는 시간 : ihour
 - 분 : min
 - 60초로 정확히 떨어지는 분 : imin
 - 초 : sec
 - 소수점 없는 초 : isec

1년의 시간



1년의 시간

```
public static void main(String[] args) {  
    final int TIME = 24;  
    final int CLOCK = 60;  
    final int PART = 60;  
    final double SUN = 365.2422;  
    int day, hour, minute, second;  
    double temp;  
  
    day = (int) SUN;  
    temp = (SUN - day) * TIME;  
    hour = (int) temp;  
    temp = (temp - hour) * CLOCK;  
    minute = (int) temp;  
    second = (int) ((temp - minute) * PART + 0.5);  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

1년의 시간

```
public static void main(String[] args) {  
    final int TIME = 24;  
    final int CLOCK = 60;  
    final int PART = 60;  
    final double SUN = 365.2422;  
    int day, hour, minute, second;  
    double temp;  
  
    day = (int) SUN;  
    temp = (SUN - day) * TIME;  
    hour = (int) temp;  
    temp = (temp - hour) * CLOCK;  
    minute = (int) temp;  
    second = (int) ((temp - minute) * PART + 0.5);  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

Light-year

- 빛(light)이 1년(365 일) 동안 가는 거리를 구하는 Program을 만들어보자



Light-year

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

■ 입력

- 빛의 속도(light) – int (300,000 Km/Sec)

- 1년(year) – int (365일)

■ 출력

- 거리(distance) – long

- 초 단위(second) – int

■ 계산 방법

- second = 1년이 몇 초인지 구함

- 거리(distance) = 초 단위(second) * 빛의 속도(light)

- 거리(distance) : 약 9조 4,600억 Km

Light-year

```
public static void main(String[] args) {  
    final int lightspeed = 300000;  
    long distance;  
  
    distance = 365 * 24 * 60 * 60 * lightspeed;  
  
    System.out.printf("빛이 1년 동안 가는 거리 : %,d Km", distance);  
}
```

결과가 이상하지 않나요?
왜 이상한 결과가 나오는지 설명하여라
바르게 고치려면

Light-year

```
public static void main(String[] args) {  
    final int YEAR = 365;  
    final int SPEED = 300000;  
    int second;  
    long distance;  
  
    second = YEAR * 24 * 60 * 60;  
    distance = (long) SPEED * second;  
  
    System.out.printf("1년은 %d 일 입니다.\n", YEAR);  
    System.out.printf("1년은 %,d 초 입니다.\n", second);  
    System.out.printf("빛의 속도 : %,d Km/Sec\n", SPEED);  
    System.out.printf("빛의 1년간 간 거리 : %,d Km\n", distance);  
  
    System.out.printf("빛이 1년 동안 가는 거리 : %,d Km", distance);  
}
```

섭씨 온도와 화씨 온도

■ 사용자로부터 섭씨 온도를 입력 받아 화씨 온도로 환산하는 Program을 작성하여라

■ 섭씨 온도 C와 화씨 온도 F사이의 관계

$$\text{화씨 온도}[^{\circ}\text{F}] = \text{섭씨 온도}[^{\circ}\text{C}] \times 9/5 + 32$$

섭씨온도와 화씨온도

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

■ 입력

- 섭씨온도 : temp (float)

■ 출력

- 화씨온도 : fahrenheit (float)

■ 계산방법

- Fahrenheit = $(9 / 5 * \text{temp}) + 32$

$$\text{섭씨온도} = (\text{화씨온도} - 32) \times 5 / 9$$

$$\text{화씨온도} = (9 / 5 \times \text{섭씨온도}) + 32$$

섭씨온도와 화씨온도

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float cdegree, fdegree;  
  
    System.out.print("섭씨온도 입력 : ");  
    cdegree = keyboard.nextFloat();  
  
    fdegree = (9.0f / 5.0f) * cdegree + 32;  
  
    System.out.printf("%.2f W2103 -> %.2f W2109Wn",  
                        cdegree, fdegree);  
}
```

섭씨온도 입력 : 12.5
12.50 °C -> 54.50 °F

섭씨온도와 화씨온도[심화]

- 변환 결과값은 소수점 셋째자리에서 반올림해서 출력하여라 (Math.round())를 사용하지 않고 처리할 것)

```
public class Main {  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        float cdegree, fdegree;  
  
        System.out.print("섭씨온도 입력 : ");  
        cdegree = keyboard.nextFloat();  
  
        fdegree = ((9.0f / 5.0f * cdegree + 32) * 100 + 0.5f) / 100;  
  
        System.out.println("섭씨 온도 : " + cdegree + "Wu2103");  
        System.out.println("화씨 온도 : " + fdegree + "Wu2109");  
    }  
}
```

화씨온도를 섭씨온도

- 화씨(°F)로 표현된 온도를 섭씨(°C)로 변환하는 프로그램을 작성하여라

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float cdegree, fdegree;  
  
    System.out.print("화씨온도 입력 : ");  
    fdegree = keyboard.nextFloat();  
  
    cdegree = (fdegree - 32) * 5.0f / 9.0f;  
  
    System.out.printf("%.2f ℥u2109 -> %.2f ℥u2103℥n",  
                        fdegree, cdegree);  
}
```

화씨온도 입력 : 50.4
50.40 °F -> 10.22 °C

Bacteria

- 한 생물학자가 새로운 Bacteria를 발견 후 1년이 지난 지금에서야 Bacteria가 1분에 20%씩 세포분열하여 성장한다는 사실을 알았다. 또한 최초의 Bacteria 세포는 10개이었음도 알았다. 현재 이 Bacteria의 세포 수는 ?

Fighter

- Mach(마하) 2의 전투기로 1년(365일) 간 거리는 ?



마하란 어떠한 속도가 음속(340 m/s)의 몇 배인가를 나타내는 단위. 즉 마하 1이면 음속과 같은 속도임

Light-year

■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

■ 입력

- 전투기의 속도(fighter) – int ($340 * 2$ m/Sec)

- 1년(year) – int (365일)

■ 출력

- 거리(distance) – long

- 초 단위(second) – int

■ 계산 방법

- second = 1년이 몇 초인지 구함

- 거리(distance) = 초 단위(second) * 전투기 속도

- 거리(distance) :

Fighter