生態模擬: 以C語言為例

Class 04 (2018/03/29)

Expression and operator

4.1 Expression and operator
4.2 Various operators
4.3 Priority of operators (運算子的優先權)
4.4 type casting (型別之轉換)

5. If statement and Loop statements
6. Application to simple models of population dynamics

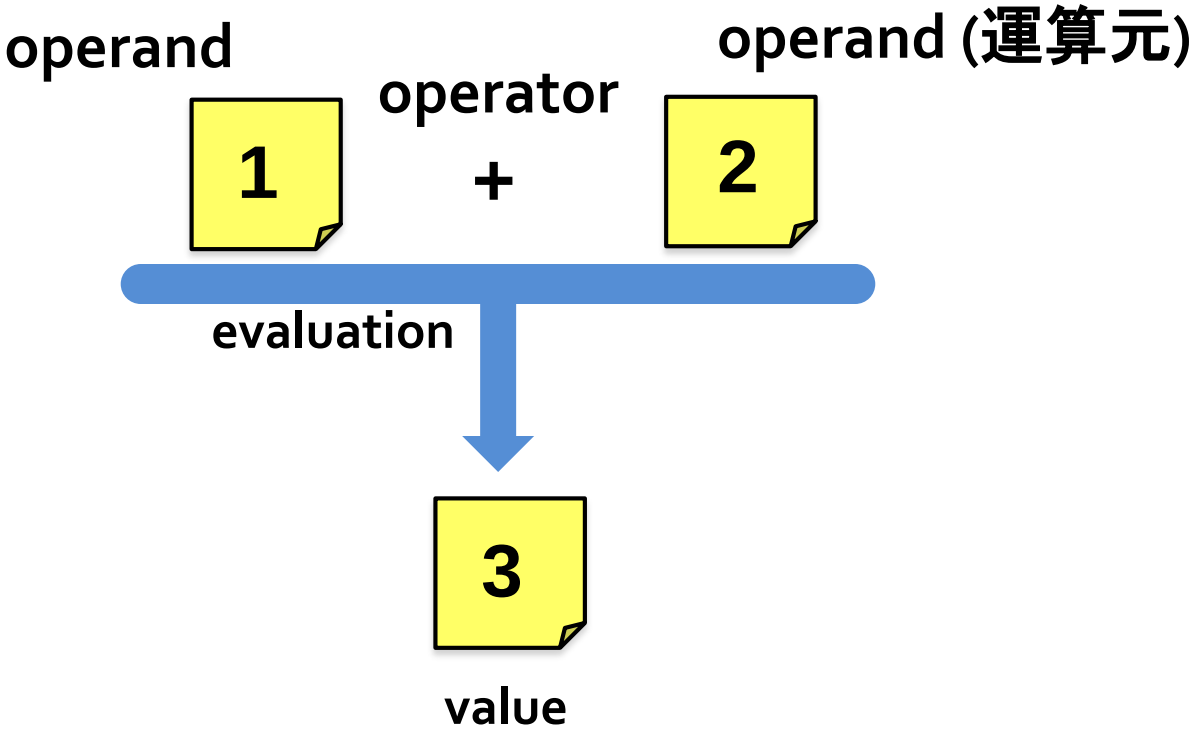
Takeshi Miki

三木　健（海洋研究所）

4.1 Expression and Operator (運算式／運算子)

What is an expression?

A combination of **operator**(s) and **operand**(s).



4.1 Expression and Operator (運算式／運算子)

How to output the value of the expression

Prepare a new sample file of C codes.

```
/**/
#include <stdio.h>

int main (void)
{

        printf("1 + 2 = %d.\n", 1 + 2);
        printf("3 x 4 = %d.\n", 3*4);

        return 0;
}
```

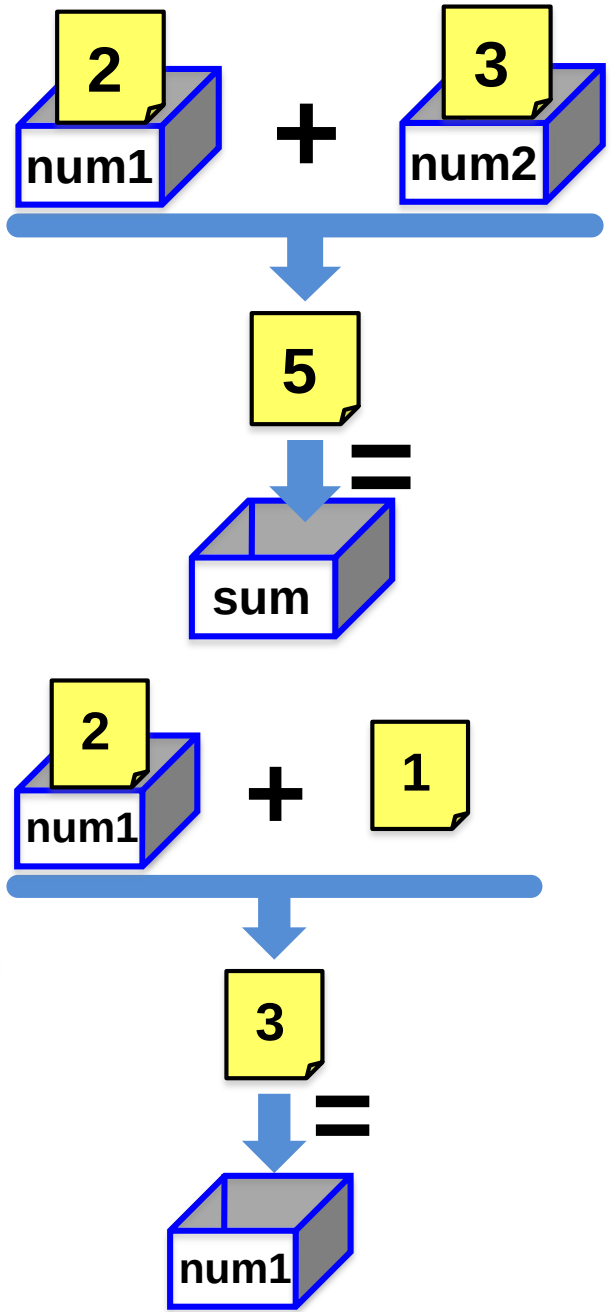
- **The evaluated value will be output on the screen.**

4.1 Expression and Operator (運算式／運算子)

Variables can be also operands.

Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
        int num1 = 2;
        int num2 = 3;
        int sum = num1 + num2;

        printf("num1 + num2 = %d.\n", sum);

        num1 = num1 + 1;
        printf("num1 + 1 = %d.\n", num1);

        return 0;
}
```



重要!!

4.2 Various operators

- Arithmetic operators (算術運算子)

+	Addition	加
-	Subtraction	減
*	Multiplication	乘
/	Division	除
%	Modulus operator	餘數

- Unary operators (一元運算子)

+	Unary Plus	positive sign 正號
-	Unary Minus	negative sign 負號

4.2 Various operators

- **Integer arithmetic**

Let x = 27 and y = 5 be 2 integer numbers.

x + y = 32
x – y = 22
x * y = 135
x / y = 5
x % y = 2

important!!

- **Floating point arithmetic**

Let x = 27.0 and y = 5.0

x + y = 32.0
x – y = 22.0
x * y = 135.0
x / y = 5.4
x % y = error!!

```
printf("num1%%num2 = %d.\n", num1%num2);
```

% 記號　轉換字元　運算子

4.2 Various operators

- Increment and decrement operators (遞增與遞減)

```
a++;
```

```
a = a + 1;
```

```
++a;
```

```
b--;
```

```
b = b - 1;
```

4.2 Various operators

- **Prefix and postfix of increment/decrement operators (前置模式與後置模式)**

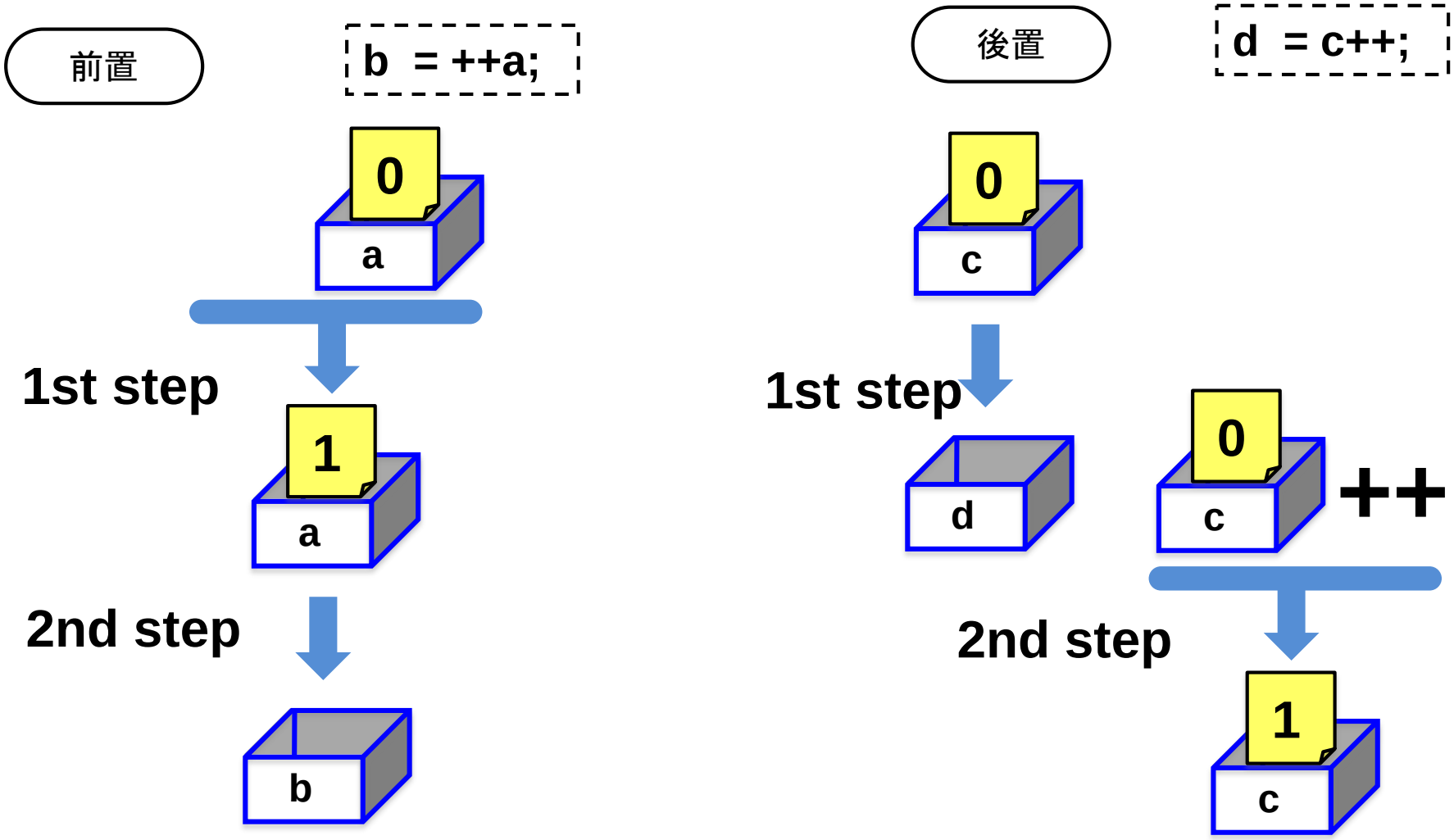
Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
        int a = 0;
        int b = 0;
        int c = 0;
        int d = 0;

        b = ++a;
        d = c++;
        printf("(a, b, c, d) = (%d, %d, %d, %d)\n", a, b, c, d);
        return 0;
}
```

前置

後置

- Prefix and postfix of increment/decrement operators (前置模式與後置模式)



4.2 Various operators

- Assignment operators (代入運算子)

=	Simple assignment	代入
+=	Addition assignment	加算代入
-=	Subtraction assignment	減算代入
*=	Multiplication assignment	乗算代入
/=	Division assignment	除算代入
	・・・	・・・

```
a += b;
```

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

```
a + = b;
```

Error!!

4.2 Various operators

- Logical operators (邏輯運算子)

&&	logical AND
||	logical OR
!	logical NOT

- Relational operators (關係運算子)

A < B	A is less than B.
A <= B	A is less than or equal to B.
A < B	A is greater than B.
A >= B	A is greater than or equal to B.
A == B	A is equal to B.
A != B	A is not equal to B.

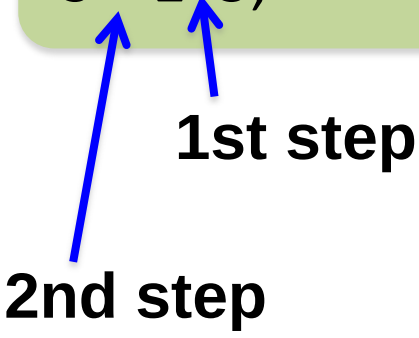
We will learn more about these two types of operators in the next week.

4.3 Priority of operators (運算子的優先權)

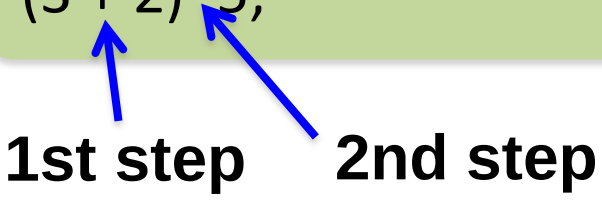
What is the priority of operators?

The simple roles are...

```
3 + 2*5;
```




```
(3 + 2)*5;
```

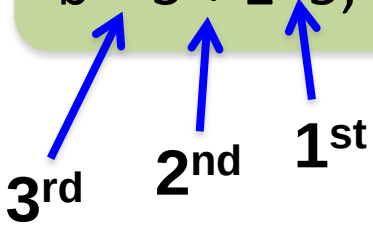


4.3 Priority of operators (運算子的優先權)

The priority of operators

Similarly,

```
b = 3 + 2*5;
```



4.3 Priority of operators (運算子的優先權)

When two operators have the same priority,

```
a + b + 1
```

From the left for an operator with “left associativity”

```
(a + b) + 1
```

```
a = b = 1
```

From the right for an operator with “right associativity”

```
a = (b = 1)
```

4.4 type casting (型別之轉換)

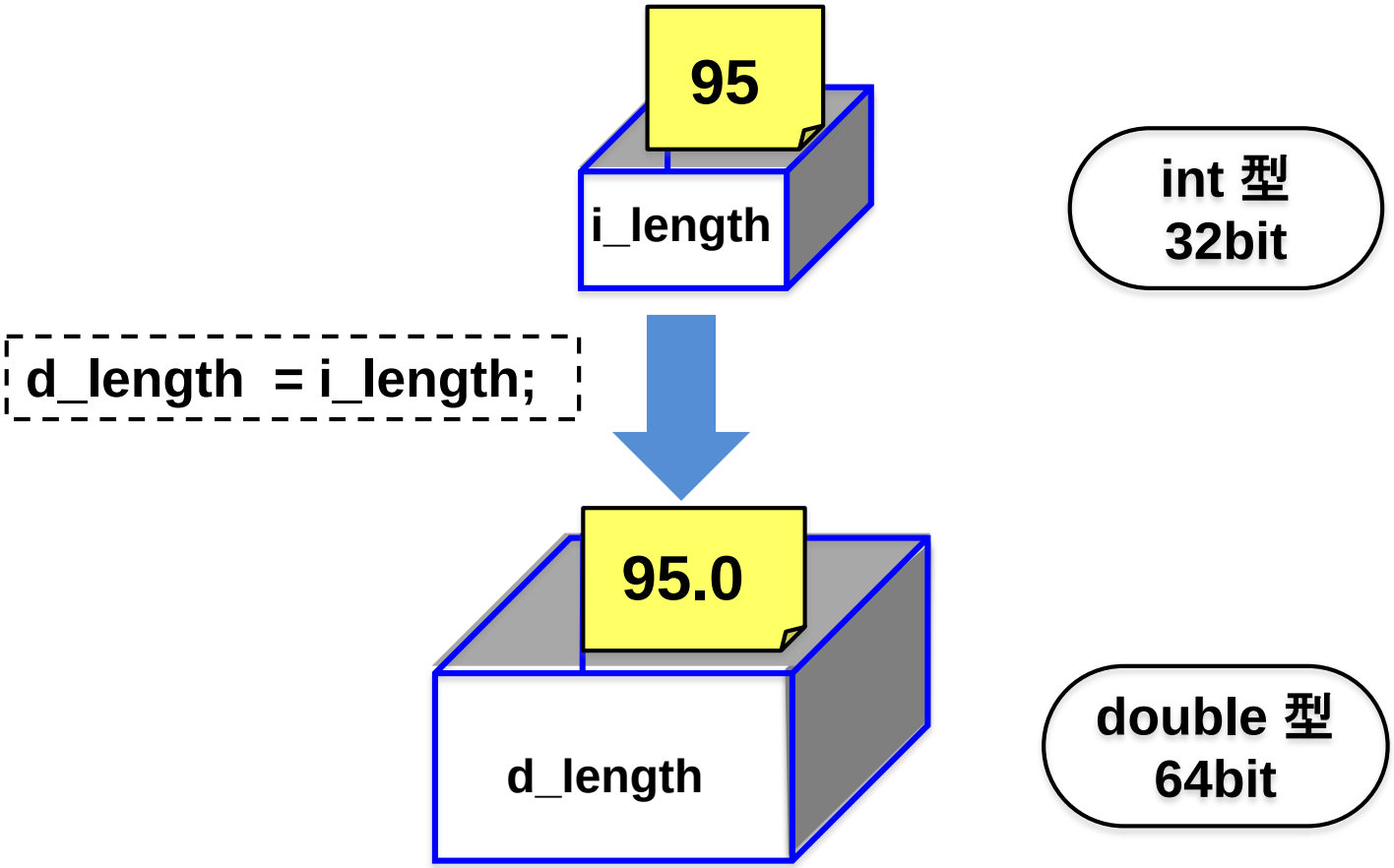
- **Assignment of a smaller variable to larger one**

Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
        int i_length = 95;
        double d_length;
        printf("The body length is %d cm.\n", i_length);

        d_length = i_length;

        printf("The body length is %lf cm.\n", d_length);

        return 0;
}
```

4.4 type casting (型別之轉換)

- Assignment of a smaller variable to larger one



4.4 type casting (型別之轉換)

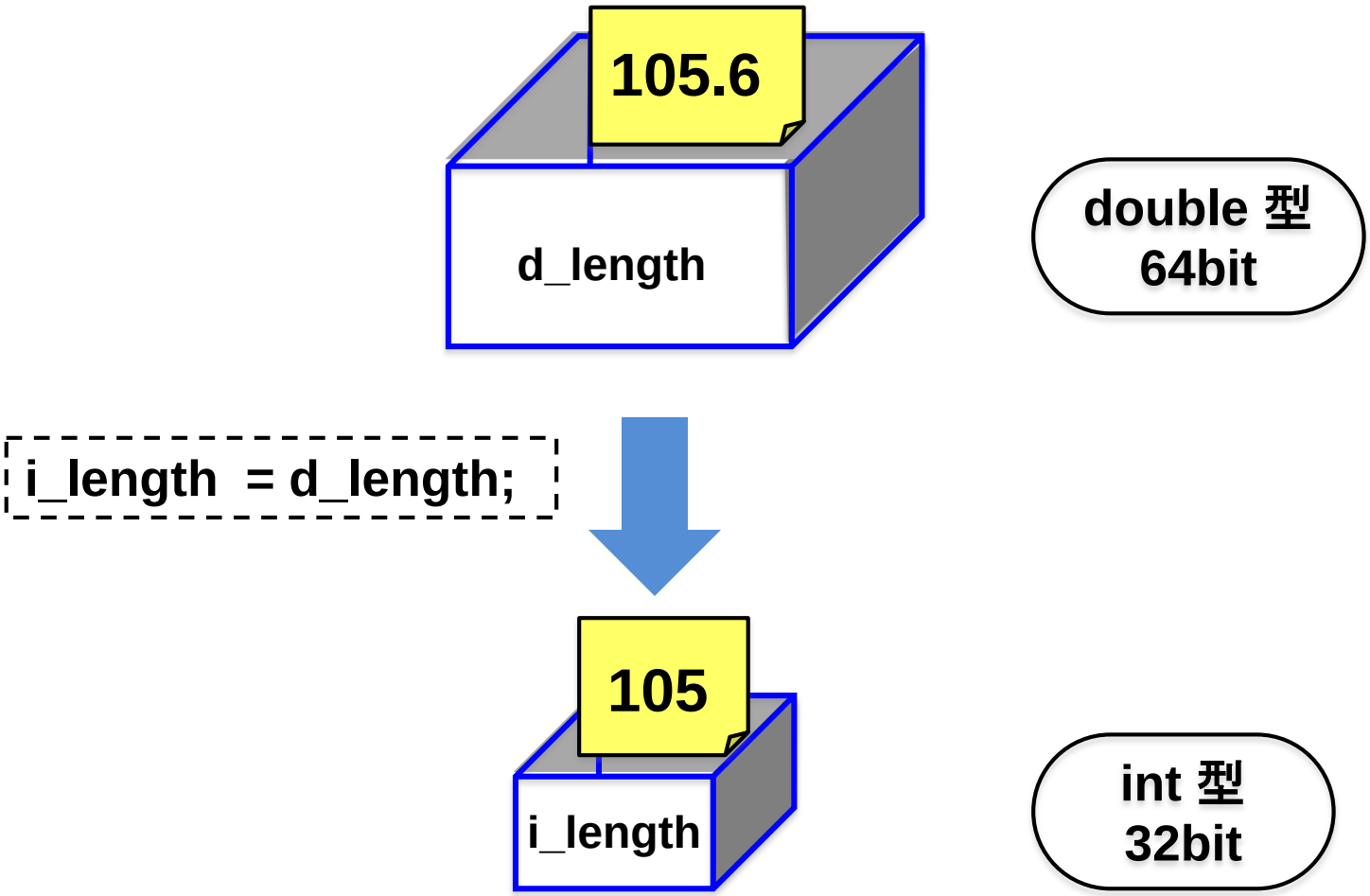
- **Assignment of a larger variable to smaller one**

Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
        int i_length;
        double d_length = 105.6;
        printf("The body length is %lf cm.\n", d_length);

        i_length = d_length;

        printf("The body length is %d cm.\n", i_length);

        return 0;
}
```

4.4 type casting (型別之轉換)

- Assignment of a larger variable to smaller one



The value will be truncated!! 重要!!

4.4 type casting (型別之轉換)

- **Maybe we cannot remember all rules for type casting. In this case, the explicit type casting would be useful.**

```
#include <stdio.h>

int main (void)
{
        int i_length;
        double d_length = 105.6;
                ...
        i_length = (int) d_length;
                ...
}
```

```
i_length = d_length;
```

4.4 type casting (型別之轉換)

- **Operator with operands of different types**

```
#include <stdio.h>

int main (void)
{
        int d = 3;
        double pi = 3.1415;

        printf("The circle with a diameter %d\n");
        printf("has the area %lf cm2. \n", (d/2.0)*(d/2.0)*pi);

        return 0;
}
```

- **The type of the operand with smaller size is first casted to the larger type.**

d/2.0 = (3.0/2.0) = 1.5

4.4 type casting (型別之轉換)

- **Operator with operands of the common types**

```
#include <stdio.h>

int main (void)
{
        int d = 3;
        double pi = 3.1415;

        printf("The circle with a diameter %d\n");
        printf("has the area %lf cm2. \n", (d/2)*(d/2)*pi);

        return 0;
}
```

- **Note: you sometimes get unexpected values !!**

重要!!

d/2 = 3/2 = 1!!

4.4 type casting (型別之轉換)

- **Type casting with the operator**

```
#include <stdio.h>

int main (void)
{
        int d1 = 3;
        int d2 = 4;
        double div1, div2;

        div1 = d1/d2;
        div2 = (double) d1/(double) d2;

        printf("div1 = %lf & div2 = %lf\n", div1, div2);

        return 0;
}
```

Homework this week

(1) Write a source code, which is intended to output/input the followings:

Please input the bottom length of a triangle (integer).	(*output*)
3 ⏎	(*input*)
Please input the height of the triangle (integer).	(*output*)
4 ⏎	(*input*)
The area of the triangle is 6.00.	(*output*)

(2) Write a source code, which is intended to output/input the followings:

Please input your height (cm) and weight (kg).	(*output*)
172.3 ⏎	(*input*)
65.0 ⏎	(*input*)
Your BMI is XXXX.	(*output*)
The standard BMI is assumed to be 24.0.	(*output*)
The difference with a standard weight is YYY kg.	(*output*)

HINT: BMI = (weight kg)/(height m)^2, and standard BMI is assumed here to be 24.0.