

# ตัวแปร ประเภทตัวแปร ตัวดำเนินการ

Variables and Operators

เอกสารประกอบการสอนวิชา CP111

มหาวิทยาลัยศรีนครินทรวิโรฒ

# เนื้อหา

## Agenda

- องค์ประกอบพื้นฐานของภาษาซี
- ตัวแปร
- ตัวดำเนินการ
- โครงโปรแกรมภาษาซี (สำหรับทบทวนความเข้าใจ)

# องค์ประกอบพื้นฐานของภาษาซี

Basic C Elements

# องค์ประกอบพื้นฐานของภาษาซี

## Basic C Elements

### □ ตัวแปร (Variables)

- เป็นชื่อที่ใช้ในการอ้างอิงถึงข้อมูลที่ถูกกำหนดประเภท
- มีลักษณะเช่นเดียวกันกับตัวแปรใดๆ ในคณิตศาสตร์ แต่มีประเภทของตัวแปรกำกับอยู่ด้วย

### □ ตัวดำเนินการ (Operators)

- เป็นการดำเนินการที่กำหนดไว้แล้วที่ใช้กระทำกับข้อมูล
- มีลักษณะเช่นเดียวกับการบวก ลบ คูณ หาร ในคณิตศาสตร์

# ตัวแปร

## Variables

# ประเภทข้อมูลพื้นฐาน

## C Data Type

### □ int

- จำนวนเต็ม (integer)

### □ double

- จำนวนทศนิยม (floating point)

### □ char

- ตัวอักษร (character)

ขนาดของตัวแปรจะขึ้นอยู่กับคอมพิวเตอร์และสถาปัตยกรรมของคอมพิวเตอร์

# ประเภทข้อมูล

## Data Type

### ☐ Integer

- short      2 bytes
- int        4 bytes
- long       4 bytes

### ☐ Floating point

- float      4 bytes
- double    8 bytes

### ☐ Character

- char       1 byte

### ☐ Others

- void

#### Optional keywords

**signed**

**unsigned**

#### Example

**unsigned short**

**unsigned int**

**signed long int**

# ชื่อตัวแปร (1/2)

## Variable Names, Identifiers

- ❑ การตั้งชื่อตัวแปรในภาษาที่จะต้องสอดคล้องกับ **context free grammar** ดังนี้

*identifier:*     *nondigit | identifier nondigit | identifier digit*

*nondigit:*     one of

*\_ a b c d e f g h i j k l m n o p q r s t u v w x y z*

*A B C D E F G H I J K L M N O P Q R S T U V W X Y Z*

*digit:*     one of

*0 1 2 3 4 5 6 7 8 9*

- ❑ สรุปสั้นๆ คือประกอบด้วยตัวอักษร **a - z, A - Z, 0 - 9** และ  
สัญลักษณ์ **\_** โดยห้ามขึ้นต้นด้วยตัวเลข **0 - 9**
- ❑ ตัวเล็กตัวใหญ่มีความแตกต่างกัน (**case sensitive**)



# ชื่อตัวแปร (2/2)

## Variable Names, Identifiers

### ❑ การตั้งชื่อที่ถูกต้อง

- `ticksPerSecond`
- `_ticksPerSecond`
- `x1`

### ❑ ตัวอย่างการตั้งชื่อที่ไม่ถูกต้อง

- `123num`
- `sur-name`
- `finished?`

ตัวพิมพ์เล็กและใหญ่แตกต่างกัน นั่นคือตัวแปรต่อไปนี้หมายถึงคนละตัวกัน

`surname`

`SurName`

`surName`

`Surname`

# การประกาศตัวแปร

## Variable Declaration

- สามารถทำได้โดยการเขียน statement ต่อไปนี้

*data-type variable-name [, variable-name, ...];*

- ตัวอย่าง

**int numStudents;**

numStudents

**double x, y;**

x

**int a = 2;**

y

a

# C Literals

## ☐ Integer

- 12345
- -12345
- 0x2A1F

## ☐ Floating point

- 1.02
- 9.18e20

## ☐ Character

- 'r'
- '\n'
- '\xA'

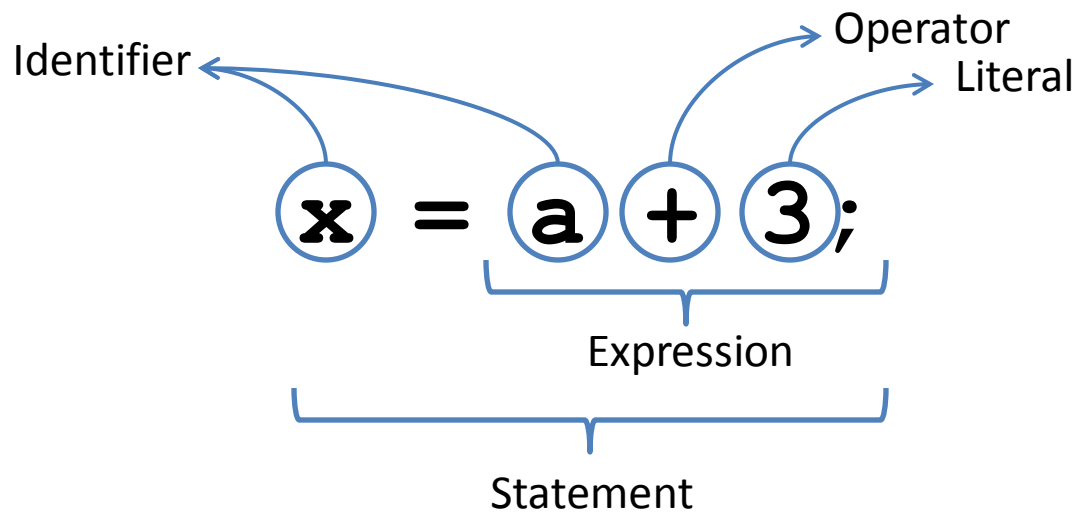
# ตัวดำเนินการ

Operators

# ตัวดำเนินการ

## Operators

- ส่วนหนึ่งเป็นตัวดำเนินการทั่วๆ ไปที่รู้จักกันดีในคณิตศาสตร์  
เช่นการบวก ลบ คูณ หาร
- ใช้ในการจัดการตัวแปร เช่น กำหนดค่า เพิ่มค่า เป็นต้น



# Expression

- ❑ เกิดจากการประกอบกันของตัวแปร (variables), ค่าคงที่ (constants), ตัวดำเนินการ (operators), และฟังก์ชัน (function calls)
- ❑ มีผลลัพธ์ของการดำเนินการเป็นข้อมูลประเภทใดประเภทหนึ่ง
- ❑ ตัวอย่าง

$y + 2$

$1 - \text{sqrt}(10)$

`count > LIMIT`

# Statement

□ ใช้แสดงการทำงานชิ้นหนึ่ง

□ ตัวอย่าง

```
a = y + 2;
```

```
b = 1 - sqrt(10);
```

```
c = 1;
```

```
;
```

# ประเภทของตัวดำเนินการ

## Classification of Operators

- ☐ Assignment Operators
- ☐ Arithmetic Operators (Additive, Multiplicative)
- ☐ Relational and Equality Operators
- ☐ Logical Operators
- ☐ Bitwise Operators
- ☐ Cast Operators

ถ้าดูตามจำนวนของตัวถูกกระทำ อาจแบ่งเป็น

- ☐ Unary operator
- ☐ Binary operator

ถ้าดูตามตำแหน่งของตัวดำเนินการ อาจแบ่งเป็น

- ☐ Infix operator
- ☐ Prefix operator
- ☐ Postfix operator



# ตัวดำเนินการกำหนดค่า

## Assignment Operators

□ ใช้ในการกำหนดค่าหรือเปลี่ยนแปลงค่าของตัวแปร

**a** **=** 5 ;

**x** **=** a + 3 ;

1. ทำการคำนวณแล้วให้ผลลัพธ์ออกมา

2. นำค่าที่ได้จากการคำนวณใน 1 มาใส่ x

# ตัวดำเนินการทางคณิตศาสตร์ (1/2)

## Arithmetic Operators

☐ การบวก

+

☐ การลบ

-

☐ การคูณ

\*

☐ การหาร

/

☐ การมอดดูโล

%

## ตัวอย่าง

☐  $5 + 3$

☐  $5 * 3$

☐  $5 \% 3$

☐  $3 + 4 + 5$

☐  $5 * 2 + 3$

- ทำจากซ้ายไปขวา (Left to right association)

- ทำ \* / ก่อน + - % (Precedence)

# ตัวดำเนินการทางคณิตศาสตร์ (2/2)

## Arithmetic Operators

- ผลลัพธ์จากการกระทำระหว่าง integer กับ floating point จะได้เป็นข้อมูลประเภท floating point

`a = 1 + 1.5;`

- เศษที่เหลือจากการหารจำนวนเต็มจะถูกตัดทิ้งไป

`a = 9 / 4;`

- ตัวดำเนินการ % หมายถึงเศษที่เหลือจากการหาร

`a = 9 % 4;`

# ตัวดำเนินการเปรียบเทียบ (1/2)

## Relational Operators

☐ มากกว่า

>

☐ มากกว่าหรือเท่ากับ

>=

☐ น้อยกว่า

<

☐ น้อยกว่าหรือเท่ากับ

<=

☐ เท่ากับ

==

☐ ไม่เท่ากับ

!=

## ตัวอย่าง

☐  $5 > 3$

☐  $5 < 3$

☐  $5 == 3$

☐  $5 != 3$

# ตัวดำเนินการเปรียบเทียบ (2/2)

## Relational Operators

□ ผลลัพธ์จากการกระทำเป็นตัวแปรประเภท `int` โดยมี  
ความหมายดังนี้

- 1            **TRUE**
- 0            **FALSE**

# ตัวดำเนินการตรรกะ (1/2)

## Logical Operators

☐ นิเสธ (NOT)

!

☐ และ (AND)

&&

☐ หรือ (OR)

||

ตัวอย่าง

☐ !TRUE

☐ TRUE && FALSE

☐ TRUE || FALSE

☐ !!!TRUE

☐ !TRUE && FALSE || TRUE

- NOT ทำจากขวาไปซ้าย  
AND, OR ทำจากซ้ายไปขวา
- ทำ NOT ก่อน AND และทำ AND ก่อน OR

# ตัวดำเนินการตรรกะ (2/2)

## Logical Operators

- ภาษาที่ไม่มีตัวแปรประเภทตรรกะ (Boolean data type)
- ภาษาซึ่งจะมีการตีความหมายตรรกะจากตัวแปรประเภทจำนวนเต็มดังนี้
  - FALSE สำหรับจำนวนเต็มศูนย์
  - TRUE สำหรับจำนวนเต็มที่ไม่ใช่ศูนย์

1

-2

-1.5

0

-1.5e10

'a'

# ตัวดำเนินการ

## Operators

`a = ! (5 > 3) && 1 + 2 != 3`



# C++ Operator Precedence

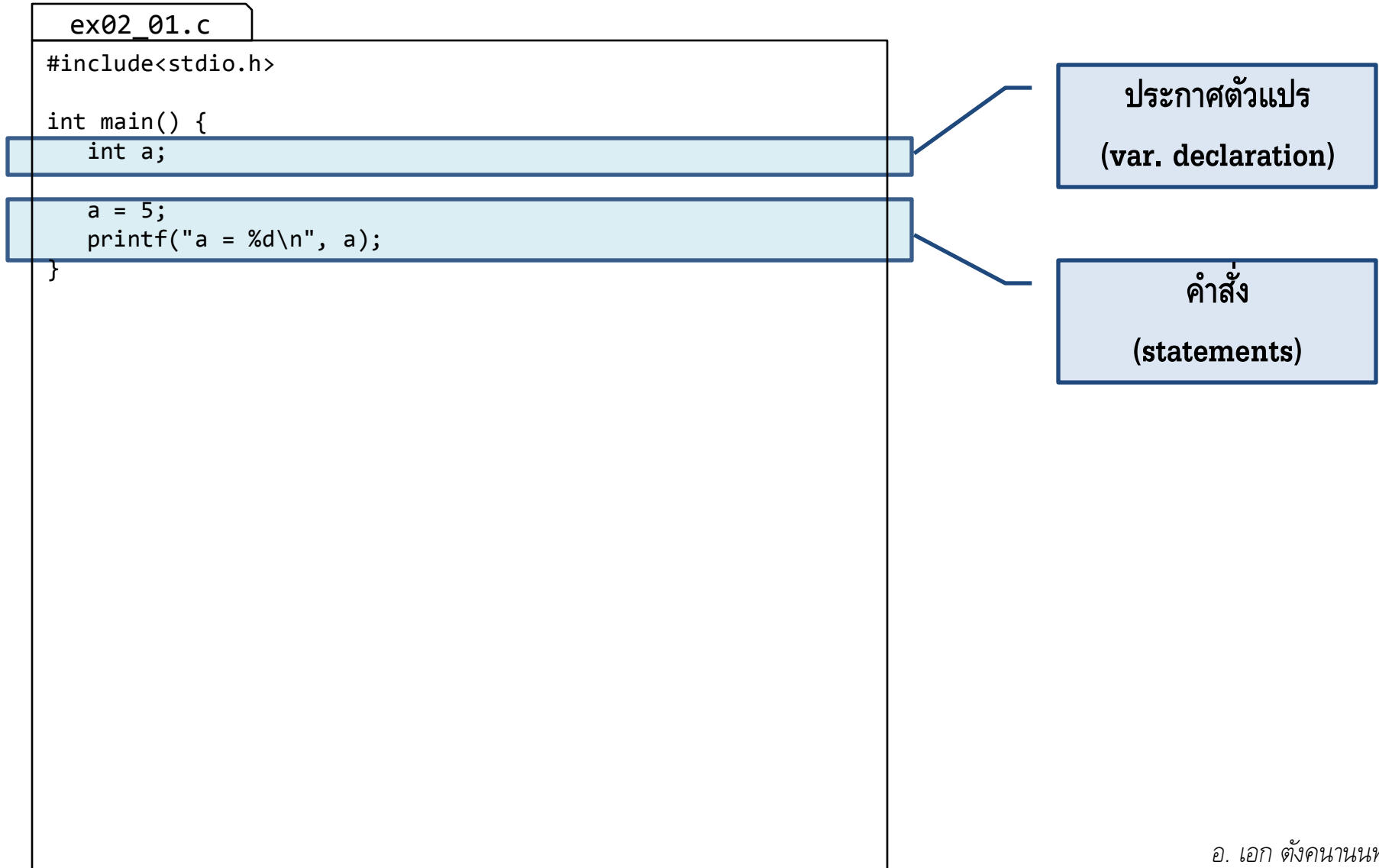
| Symbol                                    | Type of Operation               | Associativity |
|-------------------------------------------|---------------------------------|---------------|
| [ ] ( ) . -> postfix ++ and postfix —     | Expression                      | Left to right |
| prefix ++ and prefix — sizeof & * + - ~ ! | Unary                           | Right to left |
| typecasts                                 | Unary                           | Right to left |
| * / %                                     | Multiplicative                  | Left to right |
| + -                                       | Additive                        | Left to right |
| << >>                                     | Bitwise shift                   | Left to right |
| < > <= >=                                 | Relational                      | Left to right |
| == !=                                     | Equality                        | Left to right |
| &                                         | Bitwise-AND                     | Left to right |
| ^                                         | Bitwise-exclusive-OR            | Left to right |
|                                           | Bitwise-inclusive-OR            | Left to right |
| &&                                        | Logical-AND                     | Left to right |
|                                           | Logical-OR                      | Left to right |
| ? :                                       | Conditional-expression          | Right to left |
| = *= /= %=<br>+= -= <<= >>=&=<br>^=  =    | Simple and compound assignment2 | Right to left |
| ,                                         | Sequential evaluation           | Left to right |

# โครงโปรแกรมภาษาซี (สำหรับทบทวนความเข้าใจ)

C Source Code Structure (for Practical Lecture Revision)

# โครงโปรแกรมภาษาซี (1/3)

## C Source Code Structure



# โครงโปรแกรมภาษาซี (2/3)

## C Source Code Structure

ex02\_02.c

```
#include<stdio.h>

int main() {
    int a;
    int b;
    float f;
    double x, y;
    char c1, c2, c3;

    a = 5;
    printf("a = %d\n", a);

    b = a > 1;
    printf("b = %d\n", b);

    f = a + 1.2;
    printf("f = %f\n", f);

    x = y = f;
    printf("(x, y) = (%f, %f)\n", x, y);

    c1 = c2 = c3 = 'a';
    printf("c1 c2 c3 = %c %c %c\n", c1, c2, c3);

    return 0;
}
```

# โครงโปรแกรมภาษาซี (3/3)

## C Source Code Structure

ex02\_03.c

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

```
#define TRUE 1
```

```
#define FALSE 0
```

```
int main() {  
    printf("TRUE = %d\n", TRUE);  
    printf("FALSE = %d\n", FALSE);  
    printf("test1 = %d\n", TRUE && FALSE);  
    printf("test2 = %d\n", TRUE || FALSE);  
    printf("test3 = %d\n", !TRUE);  
    printf("test4 = %d\n", (1 > 2) != TRUE);  
  
    return 0;  
}
```

ประกาศมาโคร  
(macro declaration)

# การบ้าน (1/2)

## Homework

- ให้ทำความเข้าใจกับ operator ต่อไปนี้
  - Assignment =
  - Arithmetic +, -, \*, /, %
  - Logical &&, ||, !
  - Relational <, <=, >, >=, ==, !=
  - Postfix Operator ++, --
- ให้ทำความเข้าใจกับคำสั่ง printf ในขั้นพื้นฐาน

# การบ้าน (2/2)

## Homework

□ สำหรับ operator นิสิตสามารถศึกษาได้จากเว็บไซต์ดังต่อไปนี้

- <http://www.cplusplus.com/doc/tutorial/operators>
- [http://msdn.microsoft.com/en-us/library/y0c1cfax\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/y0c1cfax(VS.80).aspx)

□ นิสิตควรใช้โครงโปรแกรมก่อนหน้านี้ในการทำความเข้าใจด้วยการปฏิบัติจริง สำหรับฟังก์ชัน printf สามารถศึกษาได้จาก

- <http://www.cplusplus.com/reference/cstdio/printf>
- <http://msdn.microsoft.com/en-us/library/wc7014hz.aspx>

# คำถามท้ายบท

จะขึ้นสไลด์ท้ายชั่วโมงที่ทำการสอน



# สำหรับสัปดาห์ต่อไป

- ✓ จะใช้เวลา 30 นาทีท้ายชั่วโมงในการถามตอบ
- ✓ ถ้าไม่มีคนถามคำถามภายในเวลา 1 นาที จะทำการสอบ

# จบบทที่ 2

The End of Chapter 2

## บทที่ 3 การรับและการแสดงข้อมูล

การรับข้อมูลทางแป้นพิมพ์และการแสดงผลข้อมูลออกทางจอคอมพิวเตอร์