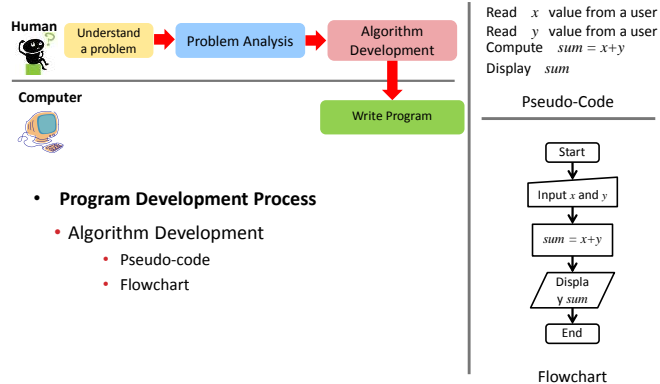
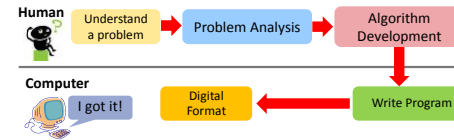


## LAST LECTURE



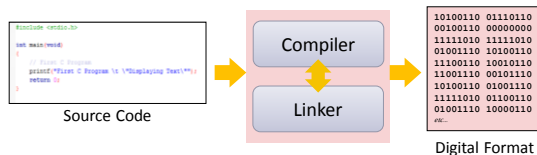
## THIS LECTURE



- Introduction to programming using C
- Element of C องค์ประกอบของภาษาซี
  - การแทนข้อมูล
  - Identifier
  - Type Casting
  - Operator in C
  - Displaying Output
    - Formatting output
  - Getting Input from a Keyboard

## INTRODUCTION TO PROGRAMMING USING C

- There are many programming languages
  - C, C++, Java, Python, etc.
  - เป็นภาษาที่มนุษย์เข้าใจ (High level language)
- คอมพิวเตอร์สามารถเข้าใจข้อมูลดิจิทัลได้เท่านั้น i.e., 0 and 1,
  - เป็นภาษาระดับต่ำ (Low level language)



- **Compilers** และ **linkers** ทำหน้าที่แปลโปรแกรมที่เขียนด้วยภาษาระดับสูง (a high level program) ให้เป็นภาษาที่คอมพิวเตอร์เข้าใจ (Low level language)

## WHY C?

### ข้อดีของภาษาซี

- ภาษาที่มีความยืดหยุ่น
- Structured language (ภาษาที่มีโครงสร้าง)
- มีคำสั่งสำเร็จรูปที่สามารถนำไปใช้ได้ เช่น **printf**, **scanf**
- ได้มีความกระชับและมีประสิทธิภาพ
- สามารถใช้ได้โนคอมพิวเตอร์หลายประเภท
- ใช้กันอย่างแพร่หลาย

## HISTORY OF C

- CPL Combined Programming Language (Barron et al., 1963)
- BCPL Basic CPL (Richards, 1969)
- B (Thompson, 1970)
- C K&R C (Ritchie, 1972)
- ANSI C American National Standards Institute C (X3J11, 1989)
- C99 (JTC1/SC22/WG14, ISO/IEC 9899, 1999)

5

## อักขรที่ใช้ในการเขียนโปรแกรมภาษาซี

- อักขระละตินตัวพิมพ์ใหญ่และพิมพ์เล็ก จำนวน 52 ตัว ได้แก่

```
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
```

ตัวพิมพ์เล็ก และ ตัวพิมพ์ใหญ่ ถือว่าเป็นอักขระคนละตัวกัน (Case sensitive)

- ตัวเลขเลขอารบิก ซึ่งเป็นเลขโดดในระบบเลขฐานสิบ จำนวน 10 ตัวเลข ได้แก่

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

- สัญลักษณ์พิเศษจำนวน 29 ตัว ดังนี้

```
! " # $ % & ' ( ) * + , - . / : ; < = > ? [ \ ] ^ _ { | } ~
```

6

## คำอ่านสัญลักษณ์พิเศษ

|                                       |                                |
|---------------------------------------|--------------------------------|
| ! exclamation mark or point           | < less than                    |
| " double quote (พินทุ, เขี้ยวคู่)     | = equal, assign                |
| # sharp or number sign, hash          | > more than                    |
| % percent                             | <> angle brackets              |
| & ampersand (and sign)                | (วงเล็บแหลม, วงเล็บสามเหลี่ยม) |
| \ single quote (เขี้ยวเดียว)          | ? Question mark                |
| ( ) round brackets, brackets (วงเล็บ) | [] square brackets             |
| * asterisk (ดอกจัน)                   | (วงเล็บเหลี่ยม, วงเล็บก้ามปู)  |
| + plus                                | ^ caret, xor                   |
| , comma                               | _ underscore                   |
| - minus, dash                         | { } curly brackets             |
| . Dot                                 | (วงเล็บปีกกา)                  |
| / slash                               | vertical bar, pipe, or         |
| \ backslash                           | ~ tilde                        |
| : colon                               |                                |
| ; semicolon                           |                                |

7

## INTRODUCTION TO PROGRAMMING USING C

```
1 #include <stdio.h>
2
3 void main()
4 {
5     // Main function
6
7     printf("Basic C Program \t Displaying Text\n");
8     /* Simple
9     C
10    Program */
11    return;
12 }
13
```

Order

#include <Header File.h>

Type Function()

{

Code

Return Value

}

Example: Source Code

- ใช้ #include เพื่อเรียกใช้คำสั่งที่มีอยู่แล้วที่ถูกกำหนดไว้ใน Header file
- printf และ scanf มีอยู่ใน stdio.h
- stdio.h ที่รวมฟังก์ชัน (คำสั่ง) สำเร็จรูปที่เรียกใช้ได้เลย เช่น ฟังก์ชันในการติดต่อกับคีย์บอร์ด จอภาพ

## INTRODUCTION TO PROGRAMMING USING C

|       |  |   |
|-------|--|---|
| Order | <pre> 1 #include &lt;stdio.h&gt; 2 3 void main() 4 { 5     // Main function 6 7     printf("Basic C Program \t Displaying Text\n"); 8     /* Simple 9      C 10     Program */ 11     return; 12 } 13 </pre> | <pre> #include &lt;Header File.h&gt;  Type Function() {     Code      Return Value } </pre> |
|-------|--|---|

Example: Source Code

ชนิด ชื่อของฟังก์ชัน และ ตัวแปรสำหรับรับค่าจากภายนอก

- void เป็นชนิดหนึ่ง
- main คือชื่อของฟังก์ชัน โดยใน 1 โปรแกรมต้องมีฟังก์ชัน main 1 ฟังก์ชัน
- () ใช้กำหนดตัวแปรสำหรับรับค่าจากภายนอกเข้ามาใช้ในโปรแกรม หากไม่กำหนด แปลว่าไม่มีรับค่าใด ๆ เข้ามา

## INTRODUCTION TO PROGRAMMING USING C

|       |  |   |
|-------|--|---|
| Order | <pre> 1 #include &lt;stdio.h&gt; 2 3 void main() 4 { 5     // Main function 6 7     printf("Basic C Program \t Displaying Text\n"); 8     /* Simple 9      C 10     Program */ 11     return; 12 } 13 </pre> | <pre> #include &lt;Header File.h&gt;  Type Function() {     Code      Return Value } </pre> |
|-------|--|---|

Example: Source Code

ใช้ { และ } เพื่อกำหนดขอบเขตของฟังก์ชัน (Function)  
ภายใน { } ใช้กำหนดคำสั่งต่าง ๆ ที่ต้องการใช้เพื่อทำงานใดงานหนึ่ง  
เช่น คำสั่ง printf ดังตัวอย่างโค้ด

## INTRODUCTION TO PROGRAMMING USING C

|       |  |   |
|-------|--|---|
| Order | <pre> 1 #include &lt;stdio.h&gt; 2 3 void main() 4 { 5     // Main function 6 7     printf("Basic C Program \t Displaying Text\n"); 8     /* Simple 9      C 10     Program */ 11     return; 12 } 13 </pre> | <pre> #include &lt;Header File.h&gt;  Type Function() {     Code      Return Value } </pre> |
|-------|--|---|

Example: Source Code

คำสั่งภายในโปรแกรม

คำสั่ง printf ดังตัวอย่างโค้ด ใช้เพื่อแสดงข้อความ

Basic C Program      Displaying text

## INTRODUCTION TO PROGRAMMING USING C

|       |  |   |
|-------|--|---|
| Order | <pre> 1 #include &lt;stdio.h&gt; 2 3 void main() 4 { 5     // Main function 6 7     printf("Basic C Program \t Displaying Text\n"); 8     /* Simple 9      C 10     Program */ 11     return; 12 } 13 </pre> | <pre> #include &lt;Header File.h&gt;  Type Function() {     Code      Return Value } </pre> |
|-------|--|---|

Example: Source Code

ต้องใช้ ; เพื่อปิดท้ายคำสั่ง

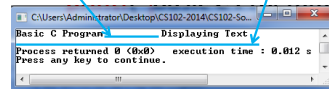
## INTRODUCTION TO PROGRAMMING USING C

การแสดงผล ข้อความ: **printf**("formatted\_Text");

**printf**("Basic C Program \t Displaying Text \n");

### Escape sequence (Example)

- \t : Tab
- \n : New line
- \a : Alert
- \\ : Displaying '\ ' character
- \" : Displaying ' ' ' character



Output with '\n'



Output without '\n'

## THIS LECTURE

- Introduction to programming using C
- Element of C องค์ประกอบของภาษาซี
  - การแทนข้อมูล
  - Identifier
  - Type Casting
  - Operator in C
  - Displaying Output
    - Formatting output
  - Getting Input from a Keyboard

## ข้อมูล — ต้องแทนในรูป BINARY CODE



### ชนิดของข้อมูลพื้นฐาน (Primitive data types)

- ตัวเลข จำนวน ปริมาณ (Numeric data) เช่น 135 หรือ 1.14125
- ตัวอักษร (Alphanumeric หรือ character data) เช่น K หรือ 3
- ข้อเท็จจริงเชิงตรรกะ (Logical data) เช่น True หรือ False

### การแทนข้อมูล

- Numeric data -> ทำเป็นเลขฐานสอง
- Alphanumeric data -> ใช้ตารางเทียบอักษรเป็นรหัสเลขฐานสอง เช่น ASCII, UNICODE, EBCDIC
- Logical data -> ใช้ 1 แทน True และ 0 แทน False

15

## VARIABLE IN C - CHARACTER


### Type Name: char

- เก็บ 1 ตัวอักษร
- ขนาดของข้อมูลคือ 1 byte (8 bits)



ASCII code

### Example :

char value = 'a';  value 0 1 1 0 0 0 0 1  
Storage of 'a' in a memory of a computer

- คอมพิวเตอร์เก็บค่าตัวอักษร 'a' ในรูปแบบเลขฐานสองโดยอ้างอิงจาก ASCII code
- ด้วยเหตุนี้ เราสามารถนำตัวอักษรมาคำนวณได้

ตัวอย่างเช่น 'a'+10 คือ

0 1 1 0 0 0 0 1  
0 0 0 0 1 0 1 0  
0 1 1 0 1 0 1 1

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | A   | B   | C  | D  | E  | F  |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|
| 0 | NUL | SOH | STX | ETX | EOT | ENQ | ACK | BEL | BS  | TAB | LF  | VT  | FF | CR | SO | SI |
| 1 | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM  | SUB | ESC | FS | GS | RS | US |
| 2 |     | !   | "   | #   | \$  | %   | &   | '   | (   | )   | *   | +   | ,  | -  | .  | /  |
| 3 | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | :   | <   | =  | >  | ?@ |    |
| 4 |     | A   | B   | C   | D   | E   | F   | G   | H   | I   | J   | K   | L  | M  | N  | O  |
| 5 |     | P   | Q   | R   | S   | T   | U   | V   | W   | X   | Y   | Z   | [  | \  | ]  | ^  |
| 6 |     | `   | a   | b   | c   | d   | e   | f   | g   | h   | i   | j   | k  | l  | m  | n  |
| 7 |     | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   | {  |    | }  | ~  |

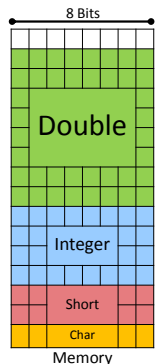
ASCII Table. Number is shown in hexadecimal.

## ชนิดของข้อมูลในภาษาซี

### • 3 ชนิด:

- Character Data ข้อมูลอักขระ
- Numerical Data ข้อมูลตัวเลข
- Logical data ข้อมูลเชิงตรรก

| Data Type | Type Name | Size (byte) |
|-----------|-----------|-------------|
| Character | char      | 1           |
| Numerical | short     | 2           |
|           | int       | 4           |
|           | long      | 4           |
|           | float     | 4           |
|           | double    | 8           |



## ข้อมูลที่ใช้ในโปรแกรม

ในการเขียนโปรแกรม ต้องเขียนรายการข้อมูลที่ต้องใช้ในการแก้ปัญหาทั้งหมด

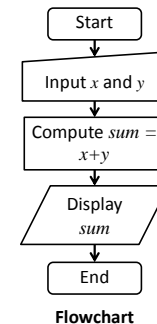
- ข้อมูลที่เป็นอินพุต (inputs)
  - ค่าที่รับจากผู้ใช้ เก็บไว้ในตัวแปร  $x$  และ  $y$
- ข้อมูลที่เป็นผลลัพธ์ (outputs)
  - ค่าที่เก็บผลรวม  $sum$
- ข้อมูลที่ใช้ระหว่างการคำนวณ

### แต่ละข้อมูลที่ใช้ต้องระบุชื่อ

- ง่ายต่อการเรียกใช้

### กำหนดชนิดของข้อมูลในภาษาซี

- ให้สอดคล้องกับชนิดของข้อมูล



Flowchart

18

## THIS LECTURE

- Introduction to programming using C
- Element of C องค์ประกอบของภาษาซี
  - การแทนข้อมูล
  - Identifier
  - Type Casting
  - Operator in C
  - Displaying Output
    - Formatting output
  - Getting Input from a Keyboard

## การระบุชื่อ (Identifier)

ชื่อ (identifiers) เพื่อระบุชื่อข้อมูลที่ใช้ในโปรแกรมภาษาซี

### 1. ชื่อค่าคงที่ (constant)

- ระบุข้อมูลที่กำหนดค่าได้เพียงครั้งเดียว และค่านั้น ๆ ไม่สามารถเปลี่ยนแปลงได้

### 2. ชื่อตัวแปร (variables)

- ระบุข้อมูลที่สามารถเปลี่ยนแปลงได้ภายในโปรแกรม
- กำหนดชนิดของข้อมูลให้แก่ชื่อ
  - ตัวเลข จำนวน ปริมาณ (Numeric data)
  - ตัวอักษร (Alphanumeric หรือ Character data)
  - ข้อมูลเชิงตรรก (Logical data)

20

## การกำหนด (ประกาศ) ชื่อค่าคงที่ในภาษา C

ใช้คำสั่ง **#define** ชื่อค่าคงที่ที่ต้องการกำหนด ค่า

กำหนดไว้นอกฟังก์ชัน

ตัวอย่าง

```
#define TRUE 1
#define FALSE 0
#define NAME_SIZE 20
#define MAX_SIZE 100
```

นิยมกำหนดชื่อค่าคงที่ให้เป็นตัวพิมพ์ใหญ่ทั้งหมด

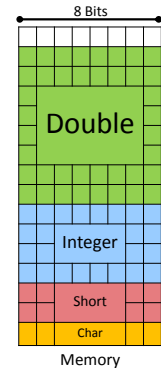
## การกำหนด (ประกาศ) ชื่อตัวแปรในภาษา C

- การประกาศตัวแปร: ทำได้ 2 แบบ

- ประกาศชนิดและชื่อของตัวแปร พร้อม กำหนดค่าเริ่มต้น
  - typeName VariableName = initialValue;**
- ประกาศชนิดและชื่อของตัวแปร
  - typeName VariableName;**

Example:

```
int num1=3; // Variable Initialize กำหนดค่าเริ่มต้น
char c = 'a';
int num2;
num2= 1; //Assign Value กำหนดค่า
double num3;
```



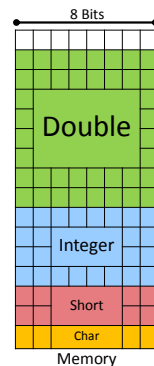
## การกำหนด (ประกาศ) ชื่อตัวแปรในภาษา C

สิ่งที่ต้องคำนึงเมื่อกำหนดตัวแปร

ต้องสร้างตัวแปรตามชนิดของข้อมูลที่แท้จริง

Example:

- ชนิดของข้อมูลคือ **จำนวนเต็ม** (integer value) หากเก็บในข้อมูลชนิด **ทศนิยม** (Double type) → สิ้นเปลืองพื้นที่
- ชนิดของข้อมูลคือ **ทศนิยม** (double value) หากเก็บในข้อมูลชนิด **จำนวนเต็ม** (Integer type) → พื้นที่เก็บข้อมูลไม่เพียงพอ



## THIS LECTURE

- Introduction to programming using C
- Element of C
  - การแทนข้อมูล
  - Identifier
  - Type Casting
  - Operator in C
  - Displaying Output
    - Formatting output
  - Getting Input from a Keyboard

## VARIABLE TYPE CASTING การแปลงชนิดข้อมูล

**Type Casting** คือวิธีการสำหรับแปลงชนิดข้อมูลจากชนิดหนึ่งไปอีกชนิดหนึ่ง

**Type Casting** ทำได้ 2 วิธี:

1. **Explicit Type Casting** การแปลงโดยตรง
  - ใช้เครื่องหมายในการแปลง
2. **Implicit Type Casting** การแปลงโดยนัย
  - Compiler แปลงชนิดของข้อมูลโดยอัตโนมัติโดยอ้างอิงจากลำดับความสำคัญของชนิด

## EXPLICIT TYPE CONVERSION (แปลงโดยตรง)

**Cast Operator :** `(type_name) expression;`

**Example1:** `double a = 20.5;`  
`int b = 10;`  
`int c;`  
`c = (int)a+b;`  
`printf("c = %d\n",c);`

Output:  
c = 30

The type of a itself is double

```
#include <stdio.h>

int main()
{
    double a = 20.5;
    int b=10, c;
    c = (int)a+b;
    printf("%d + %d = %d", a,b,c);
}
```

Code of Example1

**Example2:** `int x = 17, y = 5;`  
`double z;`  
`z = (double)x/y;`  
`printf("z = %f\n",z);`

Output:  
z = 3.400000

How about `z = (double)(x/y);` ?

Casting1.c

## IMPLICIT TYPE CONVERSION (การแปลงโดยนัย)

**2. Usual Arithmetic Conversion:** Implicitly perform to cast type of a variable according to type significance.

Compiler แปลงชนิดของข้อมูลโดยอัตโนมัติโดยอ้างอิงจาก variable type significance ของตัวแปรที่อยู่ติดกัน

| Variable Type     |  |
|-------------------|--|
| long double       | Highest Significance<br>↑<br>Lowest Significance |
| double            |  |
| float             |  |
| unsigned long int |  |
| long int          |  |
| unsigned int      |  |
| int               |  |
| short             |  |
| char              |  |
|                   |  |

## IMPLICIT TYPE CONVERSION (การแปลงโดยนัย)

| Variable Type     |                        |
|-------------------|------------------------|
| long double       | Highest<br>↑<br>Lowest |
| double            |                        |
| float             |                        |
| unsigned long int |                        |
| long int          |                        |
| unsigned int      |                        |
| int               |                        |
| short             |                        |
| char              |                        |
|                   |                        |

**Example1:** `int a = 17;`  
`char ch = 'c'; // Ascii value = 99`  
`int sum;`  
`sum = a + ch;`  
`printf("sum = %d\n",sum);`

Output:  
sum = 116

**Example2:** `int a = 17;`  
`char ch = 'c'; // Ascii value = 99`  
`double sum;`  
`sum = a + ch;`  
`printf("sum = %d\n",sum);`

Output:  
sum = 116.00000

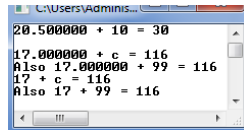
Casting2.c

## VARIABLE TYPE CASTING

```
#include <stdio.h>

int main()
{
    double a = 20.5;
    int b=10, c;
    c = (int)a+b;
    printf("%lf + %d = %d\n\n",a,b,c);

    double e = 17;
    int sum;
    char d = 'c';
    sum = e+d;
    printf("%lf + %c = %d \nAlso %lf + %d = %d\n",e,d,sum, e,d,sum);
    printf("%d + %c = %d \nAlso %d + %d = %d\n", (int)e,d,sum, (int)e,d,sum);
}
```



Casting.c

## THIS LECTURE

- Introduction to programming using C
- Element of C
  - การแทนข้อมูล
  - Identifier
  - Displaying Output
    - Formatting output
  - Type Casting
  - Operator in C
  - Getting Input from a Keyboard
- Introduction to Control Structures
  - Sequential Structure

## OPERATOR PRECEDENCE IN C

### Arithmetic Operator in C

- $A = 20+10*15/5$  ;
  - Is A = 90 or 50? } Equal?
- $B = 20+((10*15)/5)$ ;
- $C = (20+10)*15/5$ ;
- $D = ((20+10)*15)/5$ ;

## OPERATOR IN C

### Unary Operator :

- โอเปอเรเตอร์ข้างเดียวแบบใช้นำหน้า เช่น ++X มีความหมายว่าให้นำค่า X มาบวก 1 และได้กลับไปเป็น X

### Postfix Operator :

- โอเปอเรเตอร์แบบเติมหลัง เช่น X++ ให้นำค่า X มาใช้คำนวณหรือทำอะไรก่อน เมื่องานอื่นเสร็จแล้วบวกค่า X เพิ่มอีก 1 และได้ค่าใหม่เป็น X

++ : increment operator.

The statement "x++" or "++x" สำหรับเพิ่มค่า X ไป 1

ลำดับการทำงานของ "x++" และ "++x" ต่างกัน :

$y = x++$  ; ➡ 1.  $y = x$ ;  
2.  $x = x+1$ ;

$z = ++x$  ; ➡ 1.  $x = x+1$ ;  
2.  $z = x$ ;

Precedence of "++" is higher than "="



## SELF REVIEW QUESTION

- Let  $x = 3$ ;  
If  $z = x--$ ,, จงตอบคำถามต่อไปนี้:  
 $x = ?$   
 $z = ?$
- Let  $y = 2$ ;  
If  $z = --y$ ,, จงตอบคำถามต่อไปนี้:  
 $y = ?$   
 $z = ?$

## OPERATOR IN C

**Binary Operator:** โอเปอเรเตอร์แบบทำงานกับข้อมูลสองข้าง

- แบ่งเป็น Operator ที่ใช้
  - คำนวณจำนวน เช่น  $++$ ,  $--$ ,  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$
  - คำนวณระดับบิต เช่น  $\&$ (and),  $|$ (or),  $\wedge$ (xor),  $\sim$ (complement)
  - พิจารณาความเท่ากัน เปรียบเทียบเชิงตรรก ผลลัพธ์จะเป็น จริง เท็จ เช่น  $==$ ,  $!=$ ,  $<$ ,  $>$ ,  $<=$ ,  $>=$

**Ternary conditional:** โอเปอเรเตอร์เทียบเงื่อนไขแบบแบ่งสามส่วน

- เช่น  $result = a > b ? x : y$ ;

**Assignment Operator:** โอเปอเรเตอร์ที่ใช้กำหนดค่าให้ตัวแปร

- แบบกำหนดค่าทันที เช่น  $=$
- แบบนำตัวแปรมาคำนวณก่อนกำหนดค่าใหม่ให้ เช่น  $*=$ ,  $/=$ ,  $\% =$ ,  $+=$ ,  $--=$

**Comma Operator:** โอเปอเรเตอร์คอมม่า ใช้แยกส่วน

- เช่น  $i = (a += 2, a + b);$  // ค่าของ  $a+2$  หลังจากนั้นค่าของ  $a+b$  แล้วเก็บค่าในตัวแปร  $i$

**Operators' rule-of-precedence** ใช้เพื่อหลีกเลี่ยงความสับสนของลำดับการคำนวณ

- $A = 20 + 10 * 15 / 5$  ;

## OPERATOR PRECEDENCE IN C

| Operator Precedence Chart    |  |               |
|------------------------------|--|---------------|
| Operator Type                | Operator                                       | Associativity |
| Primary Expression Operators | $() [] . -> expr++ expr--$                     | left-to-right |
| Unary Operators              | $* \& + - ! ~ ++expr --expr (typecast) sizeof$ | right-to-left |
| Binary Operators             | $* / \%$                                       | left-to-right |
|                              | $+$  |               |
|                              | $-$  |               |
|                              | $>> <<$  |               |
|                              | $< > <= >=$                                    |               |
|                              | $== !=$  |               |
|                              | $\&$   |               |
|                              | $\wedge$                                       |               |
|                              | $ $  |               |
|                              | $\&\&$   |               |
|                              | $  $   |               |
| Ternary Operator             | $?:$   | right-to-left |
| Assignment Operators         | $= += -= *= /= \%> >>= <<= \&= \&\&=  $        | right-to-left |
| Comma                        | $,$  | left-to-right |

Source:  
<http://www.swansontec.com/>

- $-5 + 4 * (20 \% 6)$
- $(-5 * (192 / 3) + 7) - 4 \% 10 * 2$
- $-10 + (++a) - 7 + 3 * 2 \% 7 * 2$  (let  $a = 7$ )

## SELF REVIEW QUESTION

What is the output of the following program

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int a=10, b=2, c=3, d = 5;
6      int x = 5, y=3, z;
7      printf("a + b * c - d = %d\n", a + b * c - d);
8      printf("(a + b) * (c - d) = %d\n", (a + b) * (c - d));
9      printf("a mod c = %d\n", a%c);
10     printf("a mod c * b = %d\n", a%c * b);
11
12     z = x++;
13     printf("z = x++; x = %d, z = %d\n", x, z);
14
15     z = ++x;
16     printf("z = ++x; x = %d, z = %d\n", x, z);
17
18     z = y--;
19     printf("z = y--; y = %d, z = %d\n", y, z);
20
21     z = --y;
22     printf("z = --y; y = %d, z = %d\n", y, z);
23     return;
24 }
```

Operator.c

## THIS LECTURE

- Introduction to programming using C
- Element of C
  - การแทนข้อมูล
  - Identifier
  - Type Casting
  - Operator in C
  - Displaying Output การแสดงผลผ่านทางหน้าจอ
    - Formatting output
  - Getting Input from a Keyboard

## DISPLAYING NUMBER AND CHARACTER

แสดงข้อมูลตัวเลขและอักขระทางหน้าจอ

**printf("FormatControlString", Arguments);**

```
#include <stdio.h>
main()
{
    // Declare variables
    float price = 19.95;
    char grade = 'A';
    // Displaying result
    printf("%d of %c grade roses cost %f per %d\n", 24, grade, price, 12);
    return;
}
```

Output:  
24 of A grade roses cost 19.95000 per 12

| Type    | Conversion Character |
|---------|----------------------|
| char    | %c                   |
| string  | %s                   |
| integer | %d                   |
| float   | %f                   |
| double  | %lf                  |

## FORMATTING OUTPUT การจัดรูปแบบ

Program 1

```
#include <stdio.h>
main()
{
    //Without Field Widths
    printf("\tDrink Menu\n");
    printf("Ice \t\t%d\n", 5);
    printf("Iced Mokka \t%d\n", 95);
    printf("Smoothies \t%d\n", 125);

    return;
}
```

```

C:\Users\Administrator\Desktop\CS102-SourceCode\format...
Drink Menu
Ice          5
Iced Mokka  95
Smoothies   125
Process returned 15 (0x0F)   execution time : 0.012 s
Press any key to continue.
```

Program 2

```
#include <stdio.h>
main()
{
    //With Field Widths
    printf("\tDrink Menu\n");
    printf("Ice \t\t%3d\n", 5);
    printf("Iced Mokka \t%3d\n", 95);
    printf("Smoothies \t%3d\n", 125);

    return;
}
```

```

C:\Users\Administrator\Desktop\CS102-SourceCode\format...
Example Using widths = 3
Drink Menu
Ice          5
Iced Mokka  95
Smoothies   125
Process returned 15 (0x0F)   execution time : 0.012 s
Press any key to continue.
```

Which one do you prefer?

## FORMATTING OUTPUT การจัดรูปแบบ

- จัดข้อความแบบกำหนดความกว้างตัวอักษร และ จัดขีดขนา
  - **%widthd**
    - printf("%7d\n", 12); → \_\_\_\_12 [Total width is 7]
  - **%0widthd**
    - printf("%07d\n", 12); → 0000012 [Total width is 7]
  - **%width.floating-pointf**
    - printf("%7.2f\n", 1.2); → \_\_\_\_1.20 [Total width including "." is 7 and number of floating point is 2.]
  - **%widths**
    - printf("%7s\n", "Hello"); → \_\_H\_e\_l\_l\_o
  - **%widthc**
    - printf("%7c\n", 'C'); → \_\_\_\_C

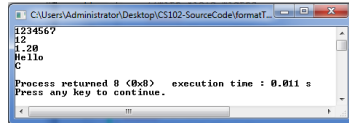
```

C:\Users\Administrator\Desktop\CS102-SourceCode\form...
1234567
12
1.20
Hello
C
Process returned 0 (0x0)   execution time : 0.000 s
Press any key to continue.
```

## FORMATTING OUTPUT การจัดรูปแบบ

- จัดข้อความแบบกำหนดความกว้างตัวอักษร และ จัดชิดซ้าย

- %-widthd**
  - `printf("%-7d\n", 12);` → 12 \_\_\_\_\_ [Total width is 7]
- %-width.floating-pointf**
  - `printf("%-7.2f\n", 1.2);` → 1.20 \_\_\_\_\_ [Total width including "." is 7 and number of floating point is 2.]
- %-widths**
  - `printf("%-7s\n", "Hello");` → H e l l o \_ \_ \_
- %-widthc**
  - `printf("%-7c\n", 'C');` → C \_ \_ \_ \_ \_



```
C:\Users\Administrator\Desktop\CS102-SourceCode\formatF...
1234567
12
1.20
Hello
C
Process returned 0 (0x0)   execution time : 0.011 s
Press any key to continue.
```

## SELF REVIEW QUESTION

### 1. เขียนคำสั่งภาษาซีที่ให้ผลลัพธ์ต่อไปนี้

- จงประกาศตัวแปรต่อไปนี้ `c`, `thisVariable`, `q76354`, และ `number` ใช้เป็นชนิด `int`
- แสดงข้อความ "This is a c program." ในหนึ่งบรรทัด
- แสดงข้อความ "This is a c program." 2 บรรทัดโดยที่บรรทัดแรกสั้นสุดที่ C
- แสดงข้อความ "This is a c program." โดยแต่ละคำอยู่คนละบรรทัด
- แสดงข้อความ "This is a c program." โดยแต่ละคำถูกแบ่งโดยใช้ Tab
- แสดงข้อความ "\_\_\_Hello CS102!"
- แสดงเลข 3.14159265359 ให้อยู่ในรูป "3.141"
- แสดงเลข 3.14159265359 ให้อยู่ในรูป "\_\_\_3.141"

### 2. จงหาและแก้ไขส่วนผิดในประโยคต่อไปนี้

- `printf("The value is %d \n", &number);`
- `printf("%d%s", number1, number2);`

## THIS LECTURE

- Introduction to programming using C
- Element of C
  - Identifier
  - Displaying Output
    - Formatting output
  - Type Casting
  - Operator in C
  - Getting Input from a Keyboard

## GETTING INPUT FROM USER การรับข้อมูลเข้าจากคีย์บอร์ด

**scanf("FormatControlString", Address\_list);**

- รับข้อมูลจากผู้ตามรูปแบบที่กำหนดไว้ใน `format-control-string`.
- ข้อมูลที่รับมาถูกเก็บใน `address_list`

ต้องใช้ & ตลอด

Example:

```
scanf("%f %d %c", &num1, &num2, &char1);
```

ผู้ใช้ต้องใส่ข้อมูล **a floating-point number** (ทศนิยม), **an integer** (จำนวนเต็ม), and **a character** (อักขระ)

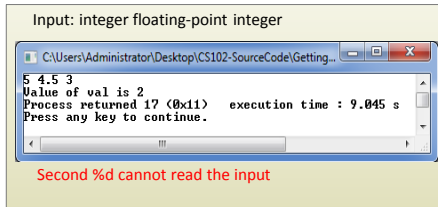
แต่ละข้อมูลถูกเก็บในตัวแปร `num1`, `num2`, และ `char1` ตามลำดับ

## GETTING INPUT FROM USER การรับข้อมูลเข้าจากคีย์บอร์ด

- การตรวจสอบว่าผู้ใช้ใส่ข้อมูลตามชนิดที่กำหนดไว้หรือไม่ทำได้โดย
- Example: `val = scanf("%d %d %d", &a, &b, &c);`

Space in between

- ถ้า `val = 3`, แปลว่า ชนิดของข้อมูลที่ใส่ทั้งหมดนั้นถูกต้อง



## SCANF - EXAMPLE 2

```
int val;  
float x, y, z;  
  
val = scanf("%f %f %f", &x, &y, &z);  
printf("%d\n", val);
```

Input: 42.5 -5 23

Output: 3

CS102 Week 5 by wdc

## SCANF -- EXAMPLE 2 (CONT)

```
int val;  
float x, y, z;  
  
val = scanf("%f %f %f", &x, &y, &z);  
printf("%d\n", val);
```

Input: 42.5 -5 c

Output: 2

CS102 Week 5 by wdc

## SCANF -- EXAMPLE 2 (CONT)

```
int val;  
float x, y, z;  
  
val = scanf("%f %f %f", &x, &y, &z);  
printf("%d\n", val);
```

Input: 42.5 c 23

Output: 1

CS102 Week 5 by wdc

**Exercise:** เขียนโปรแกรมรับความกว้าง และความยาวของสี่เหลี่ยมจากผู้ใช้ และแสดงพื้นที่ของสี่เหลี่ยม

## CONCLUSION

- องค์ประกอบของภาษาซี **Element of C**
  - ตัวระบุ ตัวแปร และ ค่าคงที่ (Identifier, Variable, Constant)
  - การแปลงชนิดข้อมูล (Type Casting)
  - Operator in C
  - การแสดงข้อมูลและค่าของข้อมูลผ่านทางหน้าจอ ( printf() )
    - การจัดรูปแบบของข้อมูลที่ใช้แสดงผล
  - การรับข้อมูลจากผู้ใช้ผ่านทางคีย์บอร์ด ( scanf() )