



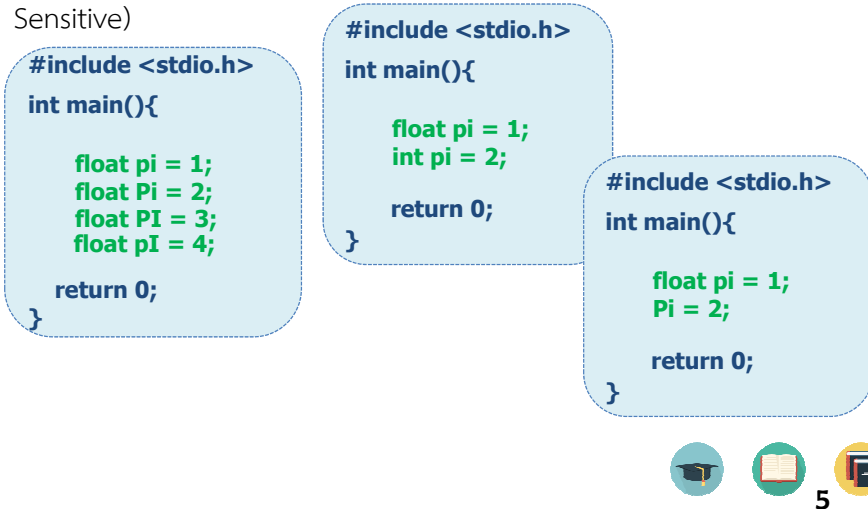
## 1.1. รูปแบบคำสั่งในภาษาซี

ภาษาซี ถือว่า ตัวอักษรพิมพ์ใหญ่ ตัวอักษรพิมพ์เล็ก เป็นคนละตัวกัน (Case Sensitive)

```
#include <stdio.h>
int main(){
    float pi = 1;
    float Pi = 2;
    float PI = 3;
    float pI = 4;
    return 0;
}
```

```
#include <stdio.h>
int main(){
    float pi = 1;
    int pi = 2;
    return 0;
}
```

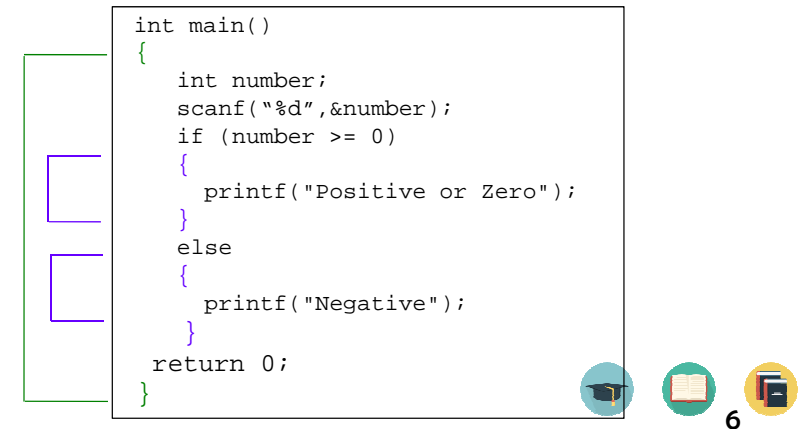
```
#include <stdio.h>
int main(){
    float pi = 1;
    Pi = 2;
    return 0;
}
```



## 1.2 บล็อก (Block)

- การเขียนโปรแกรมภาษา C จะกำหนดจุดเริ่มต้นและจุดสิ้นสุดของโปรแกรมด้วยเครื่องหมายปีกกา {...} เรียกว่า **บล็อก** ซึ่งภายในโปรแกรมก็อาจจะมีบล็อกย่อยๆ ซ้อนอยู่ข้างในได้อีกชั้นเช่น

```
int main()
{
    int number;
    scanf("%d",&number);
    if (number >= 0)
    {
        printf("Positive or Zero");
    }
    else
    {
        printf("Negative");
    }
    return 0;
}
```

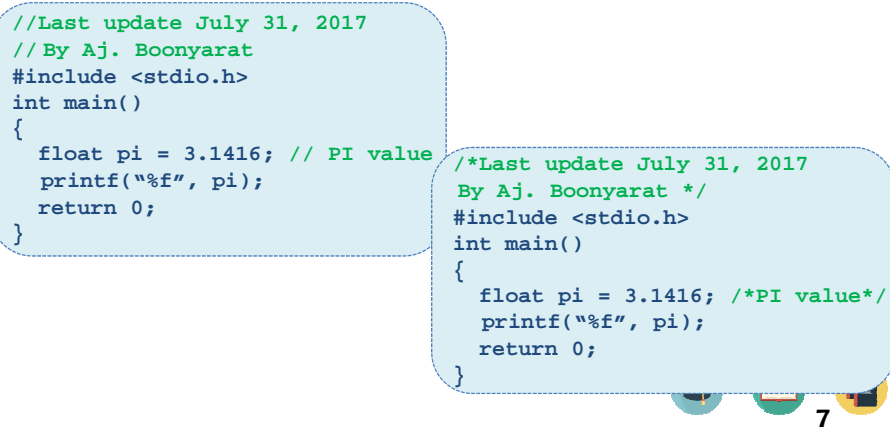


## 1.3 การเขียนคำอธิบายในภาษาซี (Comment)

- คอมเมนต์ใช้เครื่องหมาย `/* comment text */` หรือ `//`
  - คอมไพเลอร์จะไม่ประมวลผล
  - ใช้อธิบายโปรแกรมให้บุคคลอื่น หรือเพื่อความเข้าใจของตัวเอง

```
//Last update July 31, 2017
// By Aj. Boonyarat
#include <stdio.h>
int main()
{
    float pi = 3.1416; // PI value
    printf("%f", pi);
    return 0;
}
```

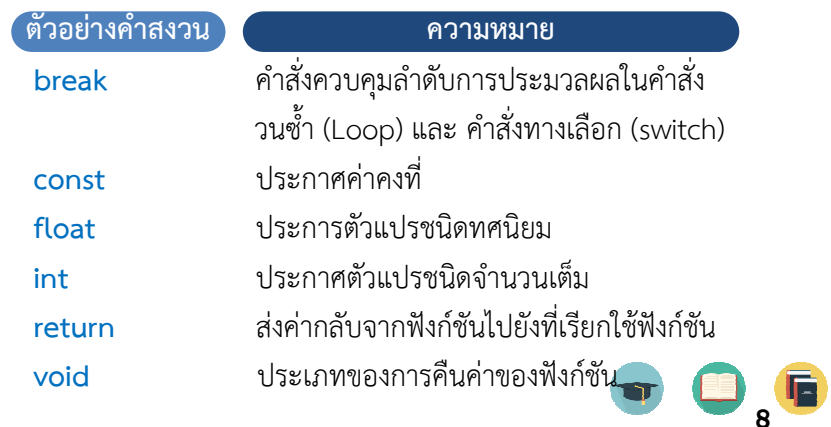
```
/*Last update July 31, 2017
By Aj. Boonyarat */
#include <stdio.h>
int main()
{
    float pi = 3.1416; /*PI value*/
    printf("%f", pi);
    return 0;
}
```



## 1.4 กลุ่มคำสั่งวนในภาษาซี

- คำสั่งวน (Reserved words or keywords) คือ คำที่ถูกกำหนดให้มีความหมายเฉพาะ และห้ามใช้เพื่อการอื่นที่ไม่ได้กำหนดเอาไว้ของคอมไพเลอร์

ตัวอย่างคำสั่งวน	ความหมาย
break	คำสั่งควบคุมลำดับการประมวลผลในคำสั่งวนซ้ำ (Loop) และ คำสั่งทางเลือก (switch)
const	ประกาศค่าคงที่
float	ประกาศตัวแปรชนิดทศนิยม
int	ประกาศตัวแปรชนิดจำนวนเต็ม
return	ส่งค่ากลับจากฟังก์ชันไปยังที่เรียกใช้ฟังก์ชัน
void	ประเภทของการคืนค่าของฟังก์ชัน



## 1.4 กลุ่มคำสงวนในภาษาซี

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



9

## 2. ชนิดข้อมูล (C Data Types)

- ชนิดข้อมูล บ่งบอกถึง รูปแบบในการเก็บข้อมูล เนื่องจากค่าของข้อมูลใดๆ จำเป็นต้องมีการกำหนดชนิด (เช่น ตัวเลข ตัวอักษร เป็นต้น) และขนาดพื้นที่หน่วยความจำ (memory) ที่ใช้ในการเก็บข้อมูล
- ชนิดข้อมูล ในภาษา C แบ่งเป็น 2 กลุ่มดังนี้
  - ชนิดข้อมูลพื้นฐาน (Primitive data types): char, int, float, double
  - ชนิดข้อมูลที่สร้างจากชนิดข้อมูลพื้นฐาน (Derived data types): Arrays, Pointers, Structures

**Note:** ภาษา C เป็นภาษาที่เรียกว่า “strongly typed” กล่าวคือ ตัวเก็บข้อมูลที่เรียกว่า ตัวแปร (variable) จะต้องถูกกำหนดชนิดข้อมูลเสมอ



10

## 2. ชนิดข้อมูล (C Data Types)

C Primitive data types: ชนิดข้อมูลพื้นฐานที่กำหนดในภาษา C

- char: ชนิดข้อมูลที่เป็นตัวอักษรขนาด 1 ตัว เช่น 'A', 'b', '1', '?'
  - int: ชนิดข้อมูลที่เป็นตัวเลขจำนวนเต็ม
  - float
  - double
- } ชนิดข้อมูลที่เป็นเลขทศนิยม

**Data Type Qualifiers:** ส่วนขยายสำหรับปรับแต่งคุณสมบัติของชนิดข้อมูล ได้แก่ ขอบเขตของค่า (Range) และขนาดพื้นที่ (size)

- Sign qualifiers: **signed, unsigned**
- Size qualifiers: **short, long**



11

## 2. ชนิดข้อมูล (C Data Types)

Data type	Size	Value range
char	1 byte	-128 to 127
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	4 byte*	-2,147,483,648 to 2,147,483,647
unsigned int	4 byte*	0 to 4,294,967,295
short int	2 byte	-32,768 to 32,767
unsigned short int	2 byte	0 to 65,535
long int	4 byte*	-2,147,483,648 to 2,147,483,647
unsigned long int	4 byte	0 to 4,294,967,295
float	4 byte	1.2E-38 to 3.4E+38
double	8 byte	2.3E-308 to 1.7E+308
long double	10 byte	3.4E-4932 to 1.1E+4932

\*32 bit compiler



12

## 2. ชนิดข้อมูล (C Data Types)

```
"D:\Course_data60\204171\lab files\ch2_code3.exe"
Minimum value for an int: -2147483648
Maximum value for an int: 2147483647
Maximum value for an unsigned int: 4294967295

Maximum value for a short int: 32767
Maximum value for an unsigned short int: 65535

Maximum value for a long int: 2147483647
Maximum value for an unsigned long int: 4294967295

Minimum value of float = 1.1754943508e-038
Maximum value of float = 3.4028234664e+038

Minimum value of double = 2.2250738585e-308
Maximum value of double = 1.7976931349e+308
```



13

## 2. ชนิดข้อมูล (C Data Types)

### ■ NOTE:

- ข้อมูลชนิด char จะมีขนาด 1 byte เสมอ
- ข้อมูลชนิด int จะมีขนาดที่เป็นไปได้ คือ 2, 4, หรือ 8 bytes ขึ้นกับเครื่องคอมพิวเตอร์ว่าเป็น 16, 32, หรือ 64 bit processor
- อย่างไรก็ตาม ความสัมพันธ์ระหว่างชนิดข้อมูล จะเป็นดังนี้เสมอ

**sizeof(short) <= sizeof(int) <= sizeof(long)**

**sizeof(float) <= sizeof(double) <= sizeof(long double)**

ALL PPT

14



### 3.1 ตัวแปร (variable)

คือ สัญลักษณ์ที่สร้างขึ้นในโปรแกรมเพื่อใช้เรียกแทนตำแหน่งของหน่วยความจำ โดยตัวแปรทุกตัวที่จะใช้งานในโปรแกรมได้จะต้องทำการประกาศก่อนเสมอ

number

10

letter

'A'

realNumber

0.732



15

### 3.1 ตัวแปร (variable)

- การประกาศตัวแปร (variable declaration) คือ การแนะนำโปรแกรมที่เขียนให้รู้จักสัญลักษณ์ที่จะใช้เก็บข้อมูล
- ตัวแปรหรือสัญลักษณ์ใหม่นั้น สามารถใช้ได้เมื่อมีการประกาศไว้ก่อน
- การประกาศตัวแปรจะต้องระบุประเภทของข้อมูล (Data type) ที่ตัวแปรนั้นจะเก็บซึ่งจะมีผลกับการจองพื้นที่ในหน่วยความจำของคอมพิวเตอร์

ไวยากรณ์การประกาศตัวแปร (Syntax)

ชนิดข้อมูล ชื่อตัวแปร;

หรือ

ชนิดข้อมูล ชื่อตัวแปร = ค่าเริ่มต้น;



16

### 3.1 ตัวแปร (variable)

#### ■ ตัวอย่างการประกาศตัวแปร

```
int    score = 99;    score (int)
char   c='A';        c (char)

char   grade;
grade = 'A';    grade (char) 'x' มาจากข้อมูลที่มีอยู่เดิมในหน่วยความจำ
```



17

### 3.1 ตัวแปร (variable)

#### กฎการตั้งชื่อตัวแปร

1. จะต้องไม่เป็นตัวอักษรพิเศษ (เช่น \*%^+&\$) และต้องต่อเนื่องกันไม่มีการเว้นช่องว่าง สามารถใช้ขีดล่าง (\_) ได้
2. อักขระตัวแรกต้องไม่ใช่ตัวเลข
3. จะต้องไม่ใช่คำสงวน (reserved words)
4. ตัวพิมพ์ใหญ่ พิมพ์เล็ก ถือว่าเป็นคนละตัว (case-sensitive)

#### ตัวอย่างการตั้งชื่อตัวแปร

S&P

H2O

2You

The student

The-student

The\_student

int

A

\_test



18

### 3.1 ตัวแปร (variable)

#### ■ Example1:

```
/* Program to Compute a triangle's area */
#include <stdio.h>
int main()
{
    float base, height, area;
    base = 3;
    height = 7;
    area = 0.5 * base * height;

    printf("The area of base %f c.m. and the height %f c.m. is %f sq.c.m.\n", base, height, area);
    return 0;
}
```

float base=3, height =7, area;  
area = 0.5 \* base \* height;

float base=3;  
float height = 7;  
float area = 0.5 \* base \* height;

```
ex "D:\Course_data60\204171\lab files\ch2_ex2.exe"
The area of base 3.000000 c.m. and the height 7.000000 c.m. is 10.500000 sq.c.m.
```



### 3.2 ค่าคงที่ (Constant)

คือ สัญลักษณ์ที่ถูกกำหนดขึ้นมา มีลักษณะคล้ายกับตัวแปร เพียงแต่ **ไม่สามารถเปลี่ยนแปลงค่าได้อีกต่อไปหลังการประกาศ**

- การกำหนดค่าคงที่เป็นชื่อนั้นจะทำให้อ่านและแก้ไขโปรแกรมได้ง่ายขึ้น
- ค่าคงที่ใช้กำหนดแทนค่าที่ใช้บ่อย ๆ ในโปรแกรม
- ค่าคงที่จะทำให้เปลี่ยนตัวเลขแค่ตำแหน่งเดียวในโปรแกรม
- การตั้งชื่อค่าคงที่ใช้กฎเดียวกับตัวแปร
  - แนวทางการเขียนโปรแกรมที่ดี ค่าคงที่จะเขียนด้วยตัวพิมพ์ใหญ่ทั้งหมด



## 3.2 ค่าคงที่ (Constant)

ไวยากรณ์การประกาศค่าคงที่(Syntax)

**const** ชนิดข้อมูล ชื่อตัวแปร = ค่าเริ่มต้น;

### ■ ตัวอย่าง การประกาศค่าคงที่

- const double PI = 3.14;
- const float TAX\_RATE = 0.07;
- const double LITER\_PER\_OZ = 0.029586;
- const double BOTTLE\_VOLUME = 2.0;  
BOTTLE\_VOLUME = 5;
- const double PI ;  
PI = 3.1416;
- const volume=9;



21

## 3.2 ค่าคงที่ (Constant)

### ■ Example 2:

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

```
int main() {
```

```
    int radius=3;
```

```
    float area, roundlen;
```

```
    area = 3.1416 * radius * radius;
```

```
    roundlen = 2 * 3.1416 * radius;
```

```
    printf("Circle area is %f\n", area);
```

```
    printf("Circumference is %f\n", roundlen);
```

```
}
```

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

```
int main() {
```

```
    int radius=3;
```

```
    float area, roundlen;
```

```
    const float PI = 3.1416;
```

```
    area = PI * radius * radius;
```

```
    roundlen = 2 * PI * radius;
```

```
    ...
```

```
}
```

ALL PPT

22



## 4. Statement

Assignment Statement

Input & Output Statement

ALL PPT

23



## 4.1 Assignment Statement

- การกำหนดค่า (assignment) ให้กับตัวแปร
- สามารถทำได้ด้วยการใช้เครื่องหมาย **assignment (=)**

ไวยากรณ์ (Syntax):

**variable = expression;**

ข้อสังเกต

- การกำหนดค่า คือการนำเอาค่าจากทางด้านขวาของเครื่องหมายเท่ากับไปเก็บไว้ทางด้านซ้ายของเครื่องหมายเท่ากับ
- ทางด้านซ้ายของเครื่องหมายเท่ากับสามารถเป็นสมการทางคณิตศาสตร์ หรือการคำนวณแบบต่าง ๆ ได้
- ทางด้านซ้ายของเครื่องหมายเท่ากับจะต้องเป็นตัวแปรเท่านั้น



24

## 4.1 Assignment Statement

### ตัวอย่างการกำหนดค่า

```
score = 9;
```

// กำหนดให้ตัวแปรชื่อ score มีค่าเท่ากับ 9

score **9**  
(int)

```
x = 3*y; // กำหนดให้ตัวแปรชื่อ x มีค่าเท่ากับ 3 เท่าของค่าที่อยู่ในตัวแปร y
```

x **45**  
(int)

y **15**  
(int)

```
double a, b, c = 5; // ประกาศ a, b, c พร้อมใส่ค่าเริ่มต้นให้ c
```

```
a = b = c*0.5; // กำหนดค่าให้ a มีค่าเท่ากับ b และ b มีค่า 0.5 เท่าของ c
```

a **2.5**  
(double)

b **2.5**  
(double)

c **5.0**  
(double)



25

## 4.2 Input & Output Statement

การสั่งให้เครื่องคอมพิวเตอร์ทำงานใดๆ และรอรับค่าผลลัพธ์จากเครื่อง จะต้องมีการติดต่อกับเครื่อง โดยการนำข้อมูลเข้าไปยังเครื่องคอมพิวเตอร์ (Input) และรับผลลัพธ์ (Output) จากเครื่องคอมพิวเตอร์

■ **Input** คือ การรับค่าข้อมูลของผู้ใช้เข้าไปในเครื่องคอมพิวเตอร์ ผ่านอุปกรณ์ ได้แก่ คีย์บอร์ด สแกนเนอร์ หรือ เมาส์

■ **Output** คือ การแสดงผลข้อความ ข้อมูล หรือ ค่าตัวแปรใดๆ ออกมาแสดงให้กับผู้ใช้ทางอุปกรณ์แสดงผลต่างๆ ได้แก่ จอภาพ ลำโพง หรือ เครื่องพิมพ์

**scanf** คำสั่งที่เชื่อมโยงกับอุปกรณ์นำข้อมูลเข้า (input) มาตรฐาน คือ คีย์บอร์ด

**printf** คำสั่งที่เชื่อมโยงกับอุปกรณ์แสดงผล (output) มาตรฐาน คือ จอภาพ



26

## 4.2 Input & Output Statement

### Output statement: printf

■ ไวยากรณ์ (Syntax):

```
printf(format specifier, argument list);
```

การใช้คำสั่ง printf ประกอบด้วย 2 ส่วนที่สำคัญคือ

- ส่วนกำหนดรูปแบบ (format specifiers): เป็นส่วนกำหนดรูปแบบการแสดงผลซึ่งจะเขียนภายใต้เครื่องหมาย “” เสมอ
- ชุดค่าที่ต้องการแสดงผล (argument list): ชุดตัวแปร ค่าคงที่ หรือ นิพจน์ที่ต้องการนำมาแสดงผล หากมีมากกว่า 1 ตัวคั่นแต่ละตัวด้วยเครื่องหมาย comma (,)



27

## 4.2 Input & Output Statement

### Output statement: printf

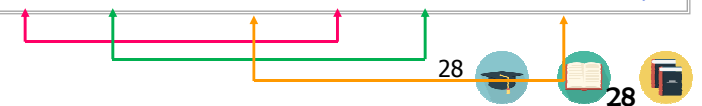
■ **printf: Format specifiers** สามารถเขียนได้ 2 ลักษณะ คือ

1. ข้อความที่ต้องการให้แสดงผลออกมา เช่น

```
printf("sum of x = ");
```

2. รหัสรูปแบบ (Format Code) ที่ใช้ในการแสดงผลซึ่งทุกรหัสรูปแบบจะต้องอยู่ตามหลังเครื่องหมาย %  
รหัสรูปแบบที่นิยมใช้ได้แก่ %c, %d, %f, %u, %o, %x, %s

```
printf("%3d %-6.0f %s", 20, 25.5, "Hello");
```



28

28



## 4.2 Input & Output Statement

### Output statement: printf

#### รหัสรูปแบบ (Format Code)

รหัสรูปแบบ	ชนิดตัวแปร	ลักษณะการแสดงผลออกจอภาพ
%d	int	ใช้แสดงข้อมูลที่เป็นเลขจำนวนเต็มฐานสิบ
%ld	long int	ใช้แสดงข้อมูลที่เป็นเลขจำนวนเต็มฐานสิบแบบ long
%u	unsigned int	ใช้แสดงข้อมูลที่เป็นเลขจำนวนเต็มฐานสิบแบบ unsigned
%c	char	ใช้แสดงข้อมูลที่เป็นตัวอักษร
%s	string	ใช้แสดงข้อมูลที่เป็นตัวแปรสตริงหรือชุดตัวอักษร
%o	int (octal)	ใช้แสดงข้อมูลที่เป็นเลขฐานแปด
%x	int (hexa)	ใช้แสดงข้อมูลที่เป็นเลขฐานสิบหก
%f	float	ใช้แสดงข้อมูลที่เป็นเลขทศนิยม ที่ไม่มีเลขยกกำลัง
%e	float, double	ใช้แสดงข้อมูลที่เป็นเลขทศนิยม ที่มีเลขชี้กำลัง
%lf	double	ใช้แสดงข้อมูลที่เป็นเลขทศนิยมแบบ double

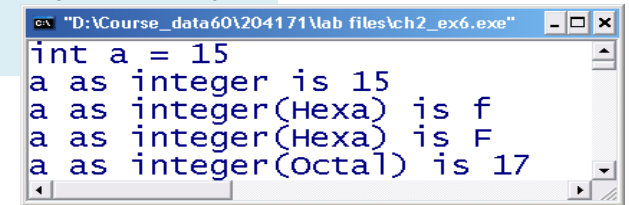
29

## 4.2 Input & Output Statement

### Output statement: printf

#### Example 3

```
#include <stdio.h> //for printf
int main()
{
    int a=15;
    printf("int a = 15 \n");
    printf("a as integer is %d \n",a);
    printf("a as integer(Hexa) is %x \n",a);
    printf("a as integer(Hexa) is %X \n",a);
    printf("a as integer(Octal) is %o \n",a);
    return 0;
}
```



30

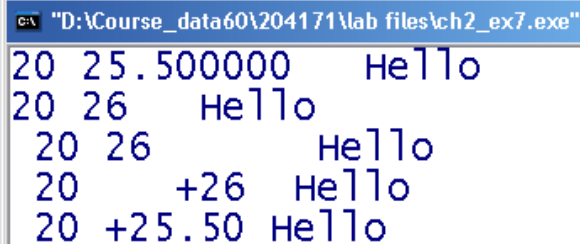
## 4.2 Input & Output Statement

### Output statement: printf

```
# include <stdio.h>
int main () {
```

#### Example 4

```
    printf("%d %f %s\n",20,25.5,"Hello");
    printf("%d %.0f %s\n",20,25.5,"Hello");
    printf("%3d %-6.0f %s\n",20,25.5,"Hello");
    printf("%3d %+6.0f %s\n",20,25.5,"Hello");
    printf("%3d %-+6.2f %s\n",20,25.5,"Hello");
    return 0;
}
```



## 4.2 Input & Output Statement

### Output statement: printf

#### Example 4 (คำอธิบาย) – การกำหนดรูปแบบตัวเลขจำนวนเต็ม และทศนิยม ยังมีเครื่องหมายเพิ่มเติมดังนี้

- เครื่องหมายลบ ให้พิมพ์ข้อมูลชิดขอบซ้าย (ปกติข้อมูลทั้งหมดจะแสดงผลชิดขวา)
- เครื่องหมายบวก ให้แสดงเครื่องหมายของค่า (ว่าเป็นค่าบวก หรือ ลบ)
- สตริงตัวเลข ระบุความกว้างของฟิลด์ (หากขนาดของตัวเลขที่แสดงผลเล็กกว่าความกว้างนี้ จะเติมช่องว่างให้ได้ขนาดตามที่ระบุ)
- จุดทศนิยม สำหรับกำหนดความกว้างของจุดทศนิยม (ค่าโดยปริยาย คือ ทศนิยม 6 ตำแหน่ง)



32



## 4.2 Input & Output Statement

### Output statement: printf

#### ■ printf: argument list

ส่วนของ argument list เป็นชุดตัวแปร ค่าคงที่ หรือ นิพจน์ที่ต้องการนำมาแสดงผล ถ้ามีมากกว่า 1 ค่าจะแยกออกจากกัน โดยใช้เครื่องหมาย Comma ( , )

หมายเหตุ ส่วน argument list นี้ไม่จำเป็นต้องมีหากส่วน **format specifier** เป็นการแสดงข้อความธรรมดา

ข้อควรระวัง ภาษาซีจะไม่มีการตรวจสอบว่าชนิดของตัวแปรที่นำค่าออก กับรหัสรูปแบบข้อมูล (Format Code) ที่กำหนดในการแสดงตรงกันหรือไม่ ดังนั้นจึงเป็นหน้าที่ของผู้เขียนโปรแกรมที่จะต้องระวังด้วยตัวเอง



33

## Quick Check1

จงจัดเรียงโปรแกรมนี้ใหม่เพื่อให้ได้โปรแกรมสามารถทำงานได้ถูกต้อง พร้อมทั้ง แสดงผลลัพธ์ของโปรแกรม

```
{
}
#include <stdio.h>
printf("Your GPA is %0.2f", GPA);
int main()
GPA=3.3351;
double GPA; // Grade Point Average
return 0;
```

ALL PPT

34



## Quick Check2:

#### ■ จงแสดงผลลัพธ์ทางจอภาพของโปรแกรมต่อไปนี้

```
#include <stdio.h>
#include <limits.h> // Library provides definitions of the characteristics of common variable types
```

```
int main()
{
    int a = 5;
    float b = 5;
    char c = 'w';
    long d = LONG_MAX; //define the maximum of long integer
    printf("a=%d\n", a);
    printf("b=%f\n", b);
    printf("c=%c\n", c);
    printf("d=%ld\n", d);
}
```



35

## 4.2 Input & Output Statement

### Input statement: scanf

#### ■ ไวยากรณ์ (Syntax):

**scanf(format code, argument list );**

การใช้คำสั่ง scanf ประกอบด้วย 2 ส่วนที่สำคัญคือ

- ส่วนกำหนดรหัสรูปแบบ (format code): เป็นส่วนกำหนดรหัสรูปแบบ(เหมือนกับรหัสรูปแบบที่ใช้ในคำสั่ง printf) ซึ่งจะเขียนภายใต้เครื่องหมาย “ ” เสมอ
- ชุดตัวแปร (argument list): ระบุตำแหน่งในหน่วยความจำของชุดตัวแปร หากมีมากกว่า 1 ตัวคั่นแต่ละตัวด้วยเครื่องหมาย comma (,)



36

## 4.2 Input & Output Statement

### Input statement: scanf

#### ■ scanf: argument list

รับค่าจากแป้นพิมพ์มาเก็บไว้ในตัวแปร และ เนื่องจากการรับข้อมูลมาเก็บในหน่วยความจำ ดังนั้นการใช้ scanf จึงต้องมีอักขระ & นำหน้าตัวแปรเสมอ

(ยกเว้น การรับข้อความ (string) จะไม่ต้องใช้เครื่องหมายนี้)

```
int num;
scanf ( "%d ", &num);
```

หมายเหตุ การใช้ scanf จะต้องมีส่วน argument list เสมอ



37

### Practice

- จงเขียนคำสั่งภาษา ประกาศตัวแปรและค่าคงที่ พร้อมทั้งกำหนดค่าตัวแปรต่อไปนี้

ข้อ	ตัวแปร/ ค่าคงที่	ชนิดตัวแปร	ชื่อตัวแปร	ค่าที่ กำหนด	คำสั่งภาษา c
1.1.	ตัวแปร	จำนวนเต็ม	number	-3	
1.2.	ตัวแปร	จำนวนจริง	grade	3.25	
1.3.	ตัวแปร	ตัวอักษร 1 ตัว	grade_level	'B'	
1.4.	ตัวแปร	จำนวนเต็ม	temp	-	
1.5.	ค่าคงที่	จำนวนจริง	PI	3.1416	
1.6.	ค่าคงที่	ตัวอักษร 1 ตัว	first_char	'A'	



## 4.2 Input & Output Statement

### Input statement: scanf

#### ■ Example 5

```
#include<stdio.h>
int main(){
    char a;
    int b;
    float c;
    printf("Enter a character:");
    scanf("%c",&a);
    printf("Enter an Integer:");
    scanf("%d",&b);
    printf("Enter a float number:");
    scanf("%f",&c);
    return 0;
}
```

Enter a character:A  
Enter an Integer:22  
Enter a float number:30.55



38

### Practice

- กำหนดให้ตัวแปร score ซึ่งเป็นตัวแปรชนิดจำนวนจริง มีค่าเท่ากับ 3.5
- เขียนคำสั่งแสดงข้อความว่า "My score is" แล้วตามด้วยค่าที่อยู่ในตัวแปร score (ทศนิยม 2 ตำแหน่ง)
- เขียนคำสั่งรับค่าตัวแปร score จากคีย์บอร์ด

