

# บทที่ 5

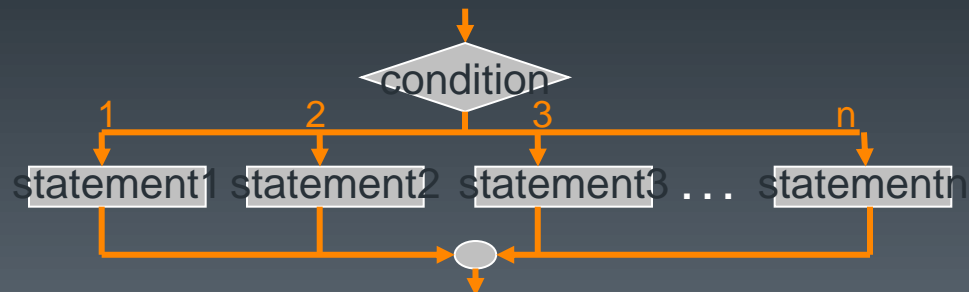
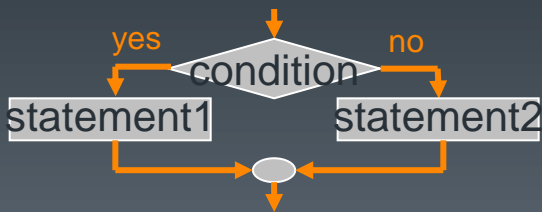
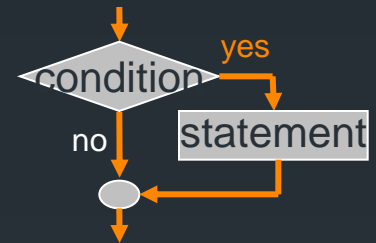
## คำสั่งเงื่อนไข

## (Conditioning Statements)

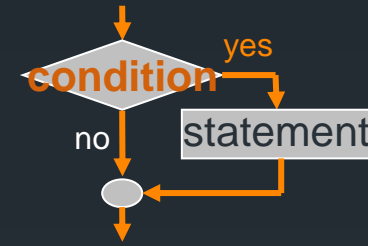
# 5.1 คำสั่งเงื่อนไข และตัวดำเนินการ

- คำสั่งเงื่อนไข ใช้ในการตัดสินใจเมื่อมีทางเลือก
- ชนิดของเงื่อนไข มี 3 ชนิด

1. เงื่อนไขทางเลือกจาก 1 ทาง (if)
2. เงื่อนไขทางเลือกจาก 2 ทาง (if-else)
3. เงื่อนไขทางเลือกจาก n ทาง (nested if, switch)



# ตัวดำเนินการ



- ตัวดำเนินการพื้นฐานในคำสั่งเงื่อนไข คือ

**ตัวดำเนินการสัมพันธ์ (Relational Operators)** เช่น

<code>==</code> (equal)	<code>!=</code> (not equal)	ถ้า $x = 75$
<code>&lt;</code> (less than),	<code>&lt;=</code> (equal & less than)	$x == 0$ ?
<code>&gt;</code> (greater than),	<code>&gt;=</code> (equal & greater than)	$x != 0$ ?
		$x <= 75$ ?
		$x > 75$ ?

- ผลจากเงื่อนไข เป็นได้ 2 แบบ คือ

- จริง (TRUE: 1) หรือ เท็จ (FALSE: 0)

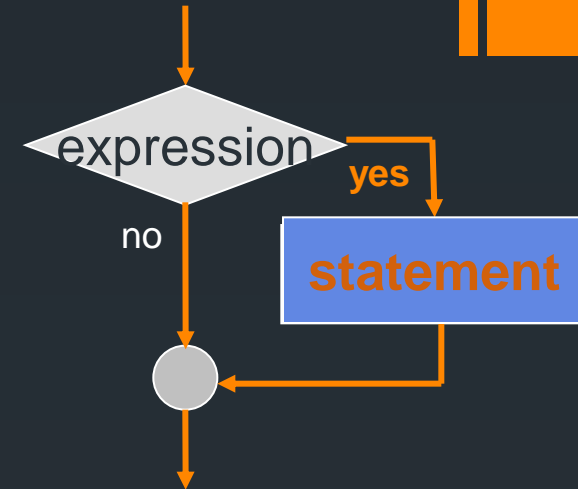
**ตัวดำเนินการตรรกะ (Logical operators)**

<code>&amp;&amp;</code> (AND)	$(x > 0) \ \&\& \ (x < 50)$ ?	
<code>  </code> (OR)	$(x > 0) \ \&\& \ (x < 90)$ ?	
<code>!</code> (NOT)	$(x < 10) \    \ (x > 90)$ ?	

# 5.2 ดำสั่ง if

## ■ รูปแบบ

```
if (expression) statement;
if (expression) { s1; s2; ... }
```

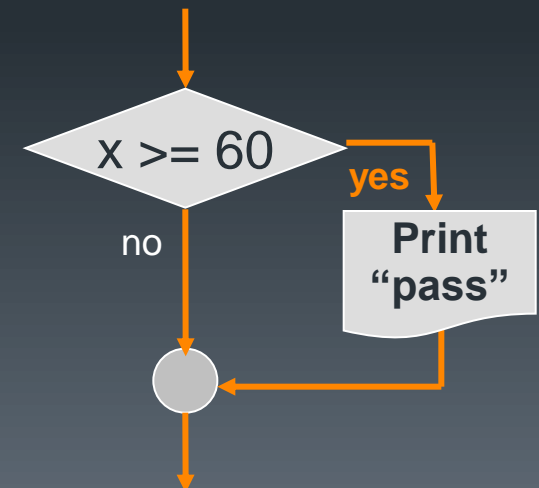


● ผลจาก **นิพจน์ตรรกะ** (Logical Expression) เป็นได้ 2 แบบคือ (TRUE หรือ FALSE)

- ถ้า **TRUE** จะทำ statement
- ถ้า **FALSE** จะไม่ทำ statement

## ● ตัวอย่างเช่น

```
if(x >= 60) printf("pass");
```



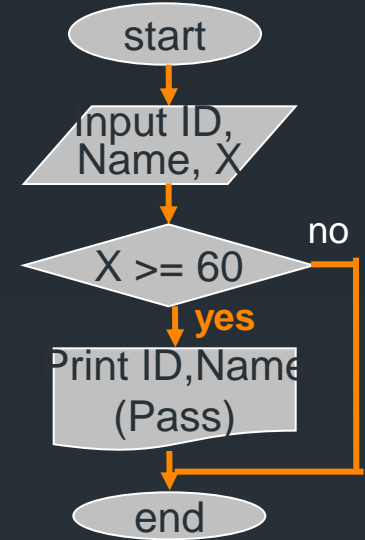
# ตัวอย่าง 5.1

เขียนโปรแกรมตรวจสอบเงื่อนไขเพื่อคัดเลือกผู้ที่สอบผ่านตามเกณฑ์ (คะแนน  $X \geq 60$ )

เมื่อ Inputs คือ ID, Name, X

## ผลลัพธ์

Enter ID): 53004100  
Enter Name: AAAAA  
Enter X (0 - 100): 75  
53004100 AAAAA (Pass)



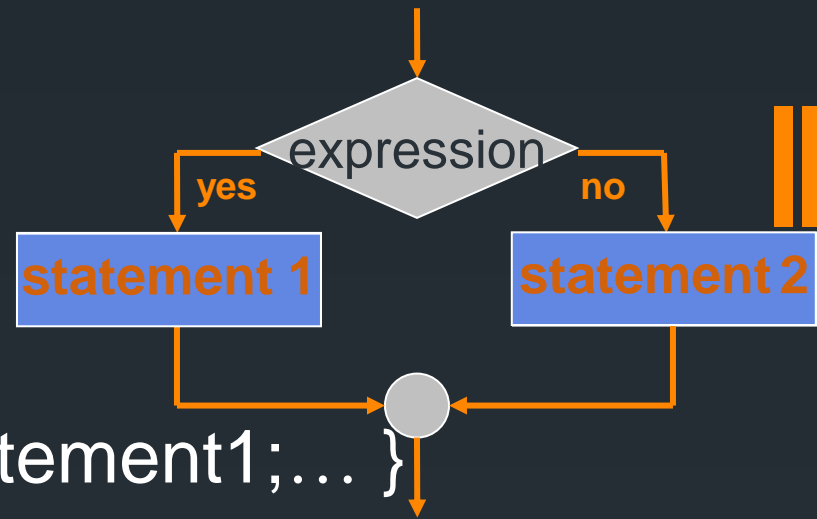
```
#include <stdio.h>
void main()
{
    int ID, X;
    char Name[20];
    printf("Enter ID: "); scanf("%d", &ID);
    printf("Enter Name: "); scanf("%s", &Name);
    printf("Enter X(0-100): "); scanf("%d", &X);

    if (X >= 60)
        printf("%d %s (Pass) \n", ID, Name);
}
```

# 5.3 ดำรง if-else

## ■ รูปแบบ

```
if (expression) { statement1; ... }  
else { statement2; ... }
```

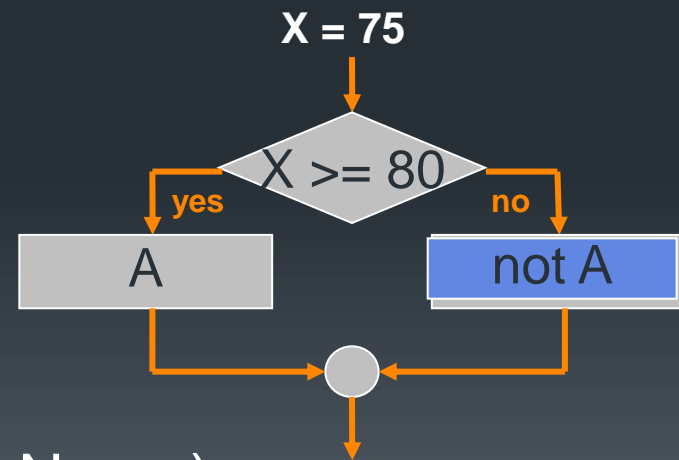


● ผลจาก นิพจน์ตรรกะ (Expression) เป็นได้ 2 แบบ คือ (TRUE หรือ FALSE)

- ถ้า TRUE จะทำ statement 1
- ถ้า FALSE จะทำ statement 2

## ● ตัวอย่างเช่น

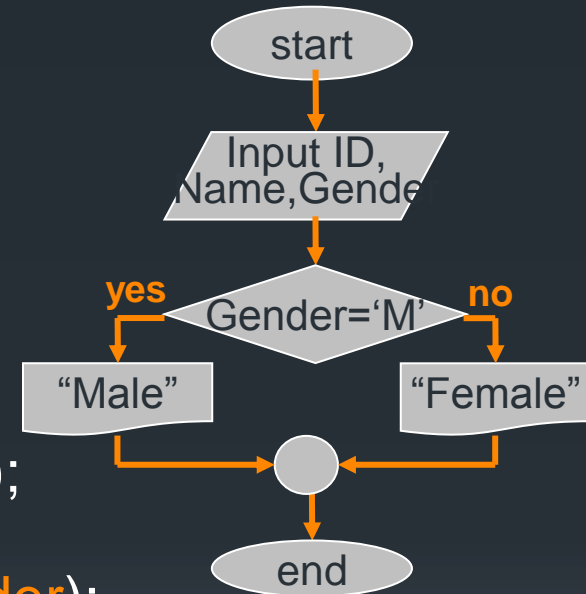
```
if (X >= 80) printf ("%d %s A\n", ID, Name);  
else printf ("%d %s not A\n", ID, Name);
```



# ตัวอย่าง 5.3

เขียนโปรแกรมตรวจสอบเงื่อนไขเพื่อแบ่งนักศึกษา ออกเป็น 2 กลุ่ม ตามเพศ (Gender) คือ เพศชาย (Male) หรือเพศหญิง (Female)

```
#include <stdio.h>
void main()
{
    int ID;
    char Name[20], Gender;
    printf("Enter ID: "); scanf("%d", &ID);
    printf("Enter Name: "); scanf("%s", &Name);
    fflush(stdin); // clear keyboard buffer
    printf("Enter Gender: "); scanf("%c", &Gender);
    if (Gender == 'M')
        printf("Male\n");
    else printf("Female\n");
}
```



## ผลลัพธ์

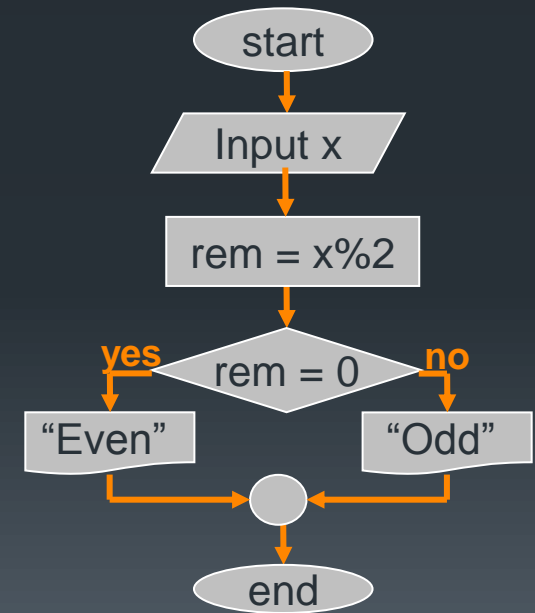
Enter ID): 53004100  
Enter Name: AAAAA  
Enter Gender: F  
Female

# ตัวอย่าง 5.4

- เขียนโปรแกรมรับค่า Integer (X) และตรวจสอบว่า X เป็นเลขคู่ (Even) หรือเลขคี่ (Odd)
- แนวคิด: X เป็น Even ถ้า X หารด้วย 2 ลงตัว

```
#include <stdio.h>
void main()
{
    int X, rem;
    printf("Enter X: "); scanf("%d", &X);
    rem = X % 2; // remainder of X/2
    if (rem == 0) printf("%d (Even)\n", X);
    else         printf("%d (odd)\n", X);
}
```

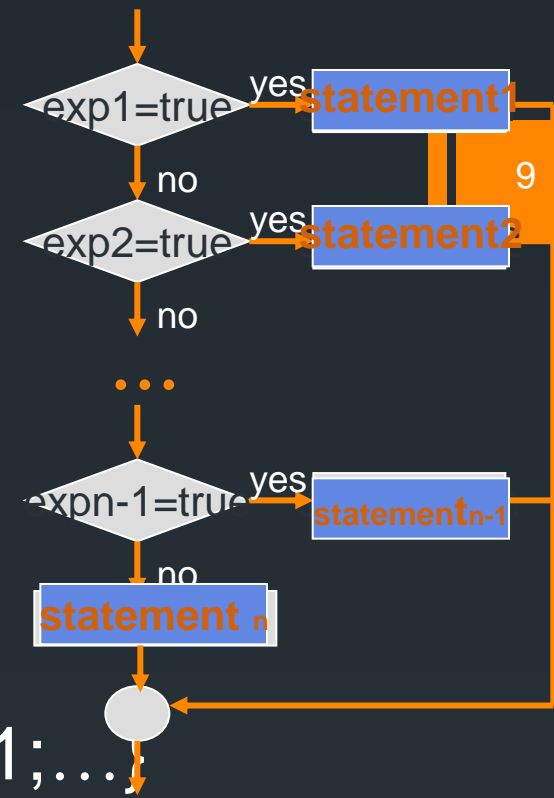
ผลลัพธ์  
Enter X: 30  
30 (Even)



# 5.4 ดำสั่ง if -else-if

## ■ รูปแบบ

```
if (exp1) {statement1;...}  
else if (exp2) {statement2;...}  
else if (exp3) {statement3;...}  
...  
else if (exp n-1) {statement n-1;...}  
else statement n;
```



● ผลจาก Expression เป็นได้ 2 แบบคือ

- ถ้า **TRUE** จะทำ **statement 1**
- ถ้า **FALSE** จะตรวจสอบ **expression 2** ซึ่งได้ผล 2 แบบ คือ
  - ถ้า **TRUE** จะทำ **statement 2**
  - ... ทดสอบเงื่อนไขต่อไปจนถึง **expression n-1**
    - ถ้า **TRUE** จะทำ **statement n-1** หรือ **FALSE** จะทำ **statement n**

# ตัวอย่าง 5.5

เขียนโปรแกรมตัดเกรดจากคะแนนสอบ (X) ตามเงื่อนไข

80 - 100 => grade = 'A'

70 - 79 => grade = 'B'

60 - 69 => grade = 'C'

50 - 59 => grade = 'D'

x < 50 => grade = 'F'

ผลลัพธ์

Enter ID: 53004100

Enter Name: AAAAA

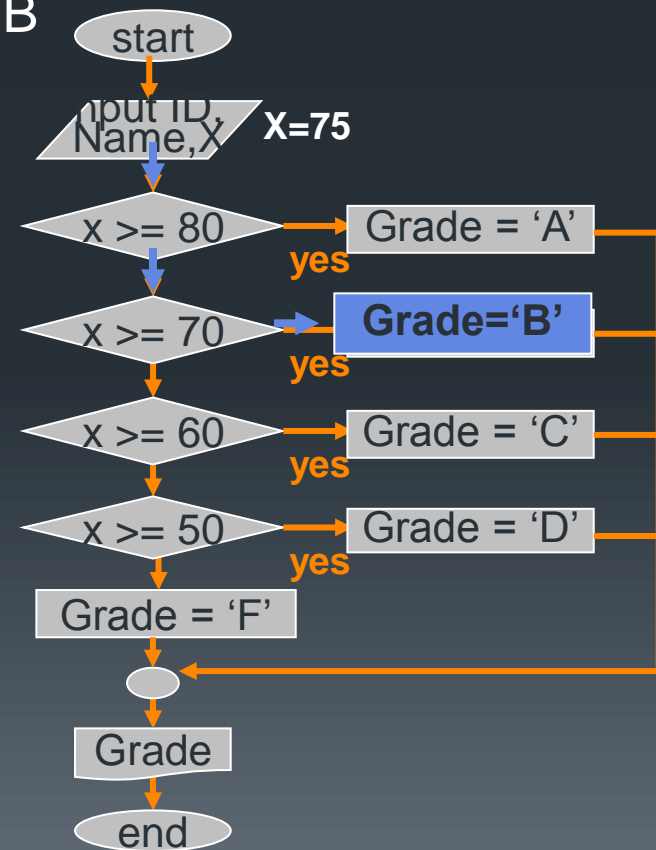
Enter X (0-100): 75

Grade = B

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

```
void main()
```

```
{    int  ID, X;
    char Name[20], Grade;
    printf("Enter ID: "); scanf("%d", &ID);
    printf("Enter Name: "); scanf("%s", &Name);
    printf("Enter X: "); scanf("%d", &X);
    if (X >= 80) Grade = 'A';
    else if (X >= 70) Grade = 'B';
    else if (X >= 60) Grade = 'C';
    else if (X >= 50) Grade = 'D';
    else Grade = 'F';
    printf("Grade = %c\n", Grade);
}
```



# ตัวอย่าง 5.6

- เขียนโปรแกรมอ่าน **Binary Expression** (+, -, \*, /)

เช่น 123.5+59.3 ที่มีรูปแบบข้อมูลเข้า คือ X Op Y

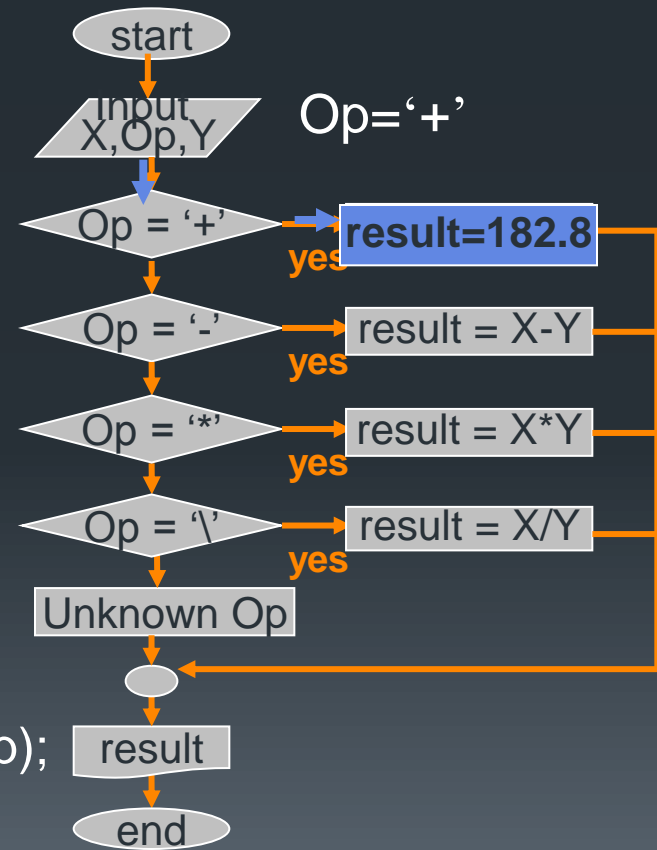
- แนวคิด:** ทดสอบค่า Op ว่าเป็น +, -, \*, หรือ /

### ผลลัพธ์

Enter ID: 123.5 + 59.3

182.80

```
#include <stdio.h>
void main()
{
    float X, Y, result=0;
    char Op;
    printf("Enter expression: ");
    scanf("%f %c %f", &X, &Op, &Y);
    if (Op == '+') result = X+Y;
    else if (Op == '-') result = X-Y;
    else if (Op == '*') result = X*Y;
    else if (Op == '/') result = X/Y; // Y > 0
    else else printf("%c \aUnknown operator\n", op);
    printf("= %.2f\n", result);
}
```



# บทที่ 6

## คำสั่งวนซ้ำ (Looping)

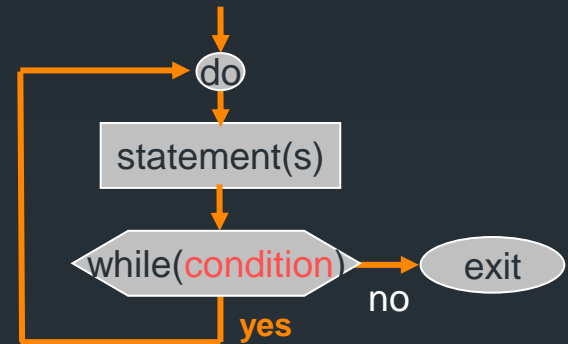
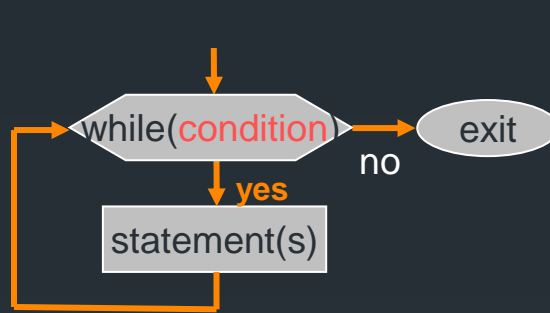
# คำสั่งวนซ้ำ

## ชนิดของการวนซ้ำ (Looping Types)

### 1. การวนซ้ำตามเงื่อนไข (Condition-controlled Loop)

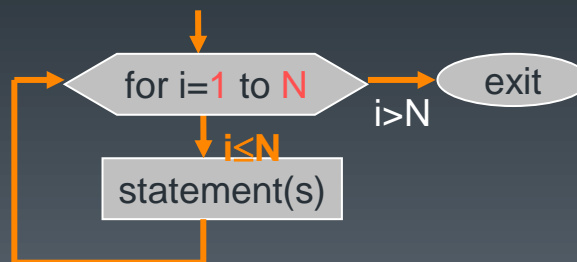
while

do-while



### 2. การวนซ้ำตามจำนวนนับ (Counting-controlled Loop)

for



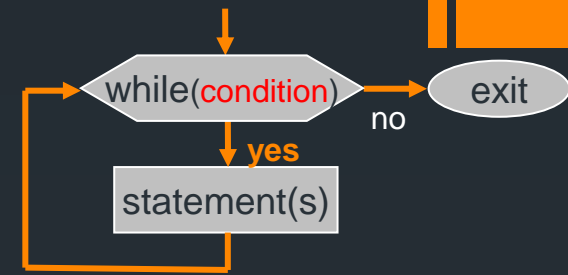
# 6.1 คำสั่ง while

## ■ รูปแบบ

**while** (expression) statement;  
or **while** (expression) {statement;...}

## ■ ถ้าเงื่อนไขใน **expression** เป็น

- **TRUE** จะทำ **statement**
- แต่ถ้า **FALSE** จะทำให้จบการทำซ้ำแบบ **while**



## ● ตัวอย่างเช่น การพิมพ์ค่า 1,2,3,4,5,6,7,8,9,10,11,12

