

ใบความรู้ที่ 9 เรื่อง โครงสร้างแบบทางเลือก *****

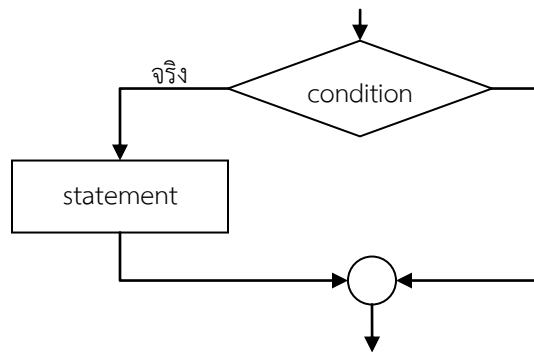
โครงสร้างแบบทางเลือก เป็นโครงสร้างที่ใช้สำหรับกำหนดทางเลือกการประมวลผลคำสั่ง โดยขึ้นอยู่กับเงื่อนไขทางเลือกที่เขียนอยู่ในรูปของนิพจน์เปรียบเทียบ (relational expression) หรือ นิพจน์ตรรกะ (boolean expression) ซึ่งใช้ใช้ตัวดำเนินการเปรียบเทียบ หรือตัวดำเนินการตรรกะเป็นตัวดำเนินการของพจน์

1. คำสั่ง if

คำสั่ง if ถูกใช้ในกรณีที่ต้องการเลือกทำเพียงทางเลือกเดียว มีรูปแบบดังต่อไปนี้

```
if ( condition )  
    statement ;
```

จากรูปแบบข้อความสั่ง if สามารถเขียนเป็นผังงานได้ตามภาพต่อไปนี้



รูปที่ 1 ผังงานข้อความสั่ง If

Condition หมายถึง เงื่อนไขที่กำหนดขึ้น เพื่อใช้พิจารณาว่าจะทำหรือไม่ทำ ผลการตรวจสอบ Condition จะได้ 2 ค่าเท่านั้นคือ จริง หรือ เท็จ โดยภาษาซีกำหนดให้ 0 มีค่าเท่ากับ เท็จ และค่าที่ไม่ใช่ 0 มีค่าเท่ากับ จริง

ตาราง ผลของการตรวจสอบ Condition

Condition	ผลการตรวจสอบ Condition
0	เท็จ
1	จริง
3	จริง
1 + 1	จริง
1 < 0	เท็จ

Statement หมายถึง คำสั่งที่ต้องการให้ทำงาน ถ้าผลการตรวจสอบ Condition เป็นจริง มี 2 ลักษณะคือ

1. **Expression Statement** เป็นการเขียนข้อความสั่งซึ่งประกอบด้วย Expression และเครื่องหมาย Semicolon (;) เช่น

```
a = 0;
a = b - c;
printf("Hello");
```

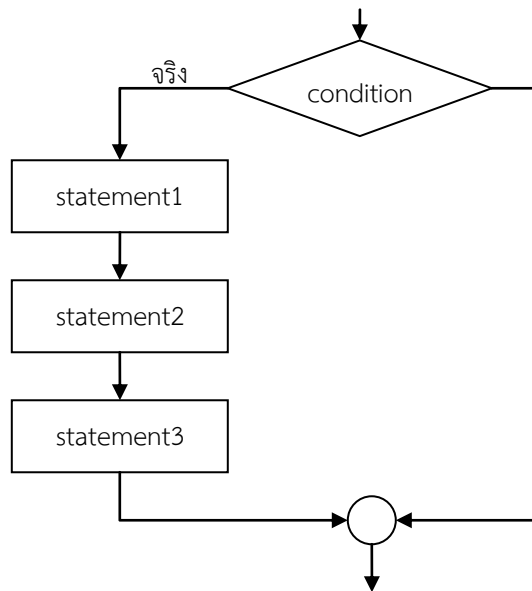
2. **Compound Statement**
คือการนำเอา Expression Statement มารวมเป็นกลุ่มเข้าด้วยกัน ภายในเครื่องหมาย { } เช่น

```
{
    a = 0;
    b = 1;
    printf("%d + %d = %d", a, b, a+b);
}
```

```
if ( condition )
{
    statement1;
    statement2;
    statement3;
}
```

จากรูปแบบของ if Statement จะมีได้เพียง 1 Statement ดังนั้นถ้าต้องการมากกว่า 1 Statement จำเป็นต้องเขียนเป็น Compound Statement เท่านั้น เช่น

สามารถเขียนเป็นผังงานได้ดังนี้



รูปที่ 2 ผังงานข้อความสั่ง if ที่มี Statement มากกว่า 1 Statement

ตัวอย่าง

โปรแกรมรับคะแนนที่ป้อนเข้ามาจากคีย์บอร์ด ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ซึ่งทำให้เงื่อนไขของข้อความสั่ง if เป็นจริง จะแสดงข้อความ Pass ปรากฏออกทางหน้าจอ

```
#include <stdio.h>
void main()
{
    int score;
    printf("Enter your score : ");
    scanf("%d", &score);
    if (score >= 50)
        printf("\nPass");
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏหน้าจอให้ป้อนคะแนนดังนี้

```
Enter your score : _
```

ถ้าป้อนคะแนนมากกว่าหรือเท่ากับ 50 ซึ่งทำให้เงื่อนไขของ if เป็นจริงจะปรากฏข้อความ Pass ออกทางหน้าจอดังนี้

```
Enter your score : 55
Pass
```

จากโปรแกรมด้านบน ถ้าเขียน Statement ต่อจาก if เช่น แสดงข้อความ Congratulation ออกทางหน้าจอ จำเป็นต้องใช้เครื่องหมาย { } เช่น

```
#include <stdio.h>
void main()
{
    int score;
    printf("Enter your score : ");
    scanf("%d", &score);
    if (score >= 50)
    {
        printf("\nPass");
        printf("Congratulation");
    }
}
```

เมื่อสร้างโปรแกรม

ถ้าป้อนคะแนน 60 จะปรากฏข้อความดังนี้

```
Enter your score : 60
Pass Congratulation
```

ถ้าป้อนคะแนน 40 จะปรากฏข้อความดังนี้

```
Enter your score : 40
```

แต่ถ้าไม่ใช่เครื่องหมาย { } จะทำให้โปรแกรมที่ได้มีผลลัพธ์การทำงานที่แตกต่างออกไป เช่น

```
#include <stdio.h>
void main()
{
    int score;
    printf("Enter your score : ");
    scanf("%d", &score);
    if (score >= 50)
        printf("\nPass");
        printf("Congratulation"); ← 1
}
```

เมื่อสร้างโปรแกรม

ถ้าป้อนคะแนน 60 จะปรากฏข้อความดังนี้

```
Enter your score : 60
Pass Congratulation
```

ถ้าป้อนคะแนน 40 จะปรากฏข้อความดังรูป

```
Enter your score : 40 Congratulation
```

1

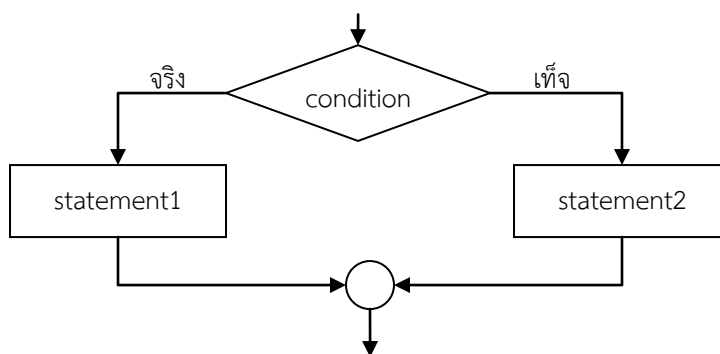
printf("Congratulation"); ไม่ได้อยู่ภายในข้อความสั่ง if ดังนั้นไม่ว่าเงื่อนไขของ if จะเป็นเช่นใด printf("Congratulation"); จะทำงานอย่างแน่นอน ซึ่งเป็นจุดที่ต้องระมัดระวังเป็นอย่างยิ่ง

2. คำสั่ง if-else

คำสั่ง if-else ถูกใช้ในกรณีที่ต้องการเลือกจากสองทางเลือก โดยใช้ผลจากการตรวจสอบเงื่อนไข มีรูปแบบดังต่อไปนี้

```
if ( condition )
    statement1 ;
else
    statement2 ;
```

รูปแบบของข้อความสั่ง if-else สามารถเขียนเป็นผังงานได้ดังนี้



รูปที่ 1 ผังงานข้อความสั่ง if-else

การทำงานจะเริ่มจากการตรวจสอบ Condition ถ้าเป็นจริงจะกระทำ Statement1 แต่ถ้าเป็นเท็จจะกระทำ Statement2 และเช่นเดียวกัน Statement1 ที่ตามหลัง if และ Statement2 ที่ตามหลัง else จะต้องเป็น Statement เดียวเท่านั้น

ตัวอย่างโปรแกรมที่ 3

โปรแกรมรับคะแนนที่ป้อนเข้ามาจากคีย์บอร์ด ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ซึ่งทำให้เงื่อนไขของข้อความสั่ง if เป็นจริง จะแสดงข้อความ Pass แต่ถ้าเงื่อนไขของข้อความสั่ง if เป็นเท็จ จะแสดงข้อความ Fail ปรากฏออกทางหน้าจอ

```
#include <stdio.h>
void main()
{
    int score;
    printf("Enter your score : ");
    scanf("%d", &score);
    if (score >= 50)
        printf("Pass\n");
    else
        printf("Fail\n");
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏหน้าจอให้ป้อนคะแนนดังนี้

```
Enter your score : _
```

ถ้าป้อนคะแนนมากกว่าหรือเท่ากับ 50 ซึ่งทำให้เงื่อนไขของ if เป็นจริงจะปรากฏข้อความ Pass ออกทางหน้าจอภาพดังนี้

```
Enter your score : 60  
Pass
```

ถ้าป้อนคะแนนน้อยกว่า 50 ซึ่งทำให้เงื่อนไขของ if เป็นเท็จจะปรากฏข้อความ Fail ออกทางหน้าจอภาพดังนี้

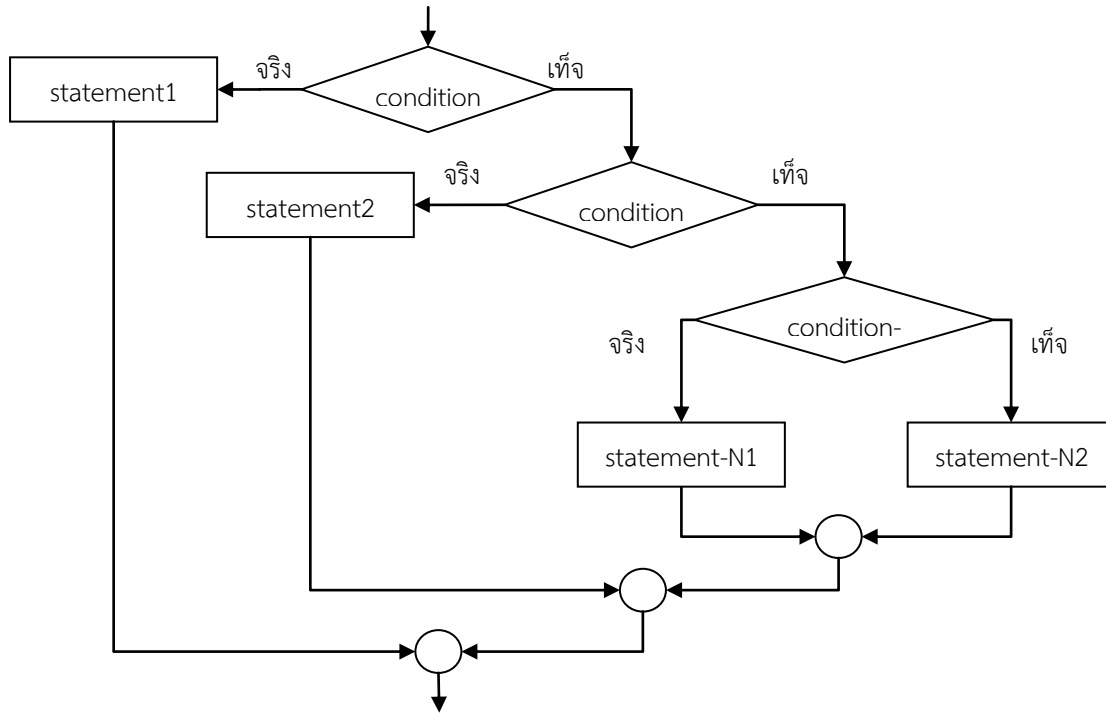
```
Enter your score : 40  
Fail
```

3. คำสั่ง else-if

คำสั่ง else-if ถูกใช้ในกรณีที่ต้องการเลือกจากทางเลือกมากกว่า 2 ทางเลือก โดยใช้ผลการตรวจสอบเงื่อนไขที่ต่างกัน มีรูปแบบดังต่อไปนี้

```
if ( condition1 )  
    statement1 ;  
else if ( condition2 )  
    statement2 ;  
...  
else if ( condition-N )  
    statement-N1 ;  
else  
    statement-N2 ;
```

รูปแบบของข้อความสั่ง else-if สามารถเขียนเป็นผังงานได้ดังนี้



รูปที่ 1 ผังงานข้อความสั่ง else-if

การทำงานจะเริ่มจากการตรวจสอบเงื่อนไขของ if แรก ถ้าผลที่ได้เป็นจริงก็จะทำ Statement1 แล้วออกจากข้อความสั่ง else-if แต่ถ้าเป็นเท็จจะตรวจสอบเงื่อนไขของ if ตัวที่สอง ถ้าผลที่ได้เป็นจริงก็จะทำ Statement2 แล้วออกจากข้อความสั่ง else-if แต่ถ้าเป็นเท็จจะตรวจสอบเงื่อนไข if ตัวต่อไปเรื่อยๆ จนกระทั่งถึง if ตัวสุดท้าย ถ้าผลที่ได้เป็นจริงก็จะทำ Statement-N1 แต่ถ้าเป็นเท็จก็จะทำ Statement-N2

ข้อสังเกต ถ้าเงื่อนไข if ตัวแรกเป็นจริง จะไม่ทำงาน if ตัวถัดไปอย่างแน่นอน ถึงแม้ว่าเงื่อนไขของ if ตัวถัดไปจะเป็นจริงก็ตาม

ตัวอย่างโปรแกรมที่ 4

โปรแกรมแสดงชื่อวันตามลำดับตัวเลขที่ป้อนเข้ามาตามตารางต่อไปนี้

ลำดับ	ชื่อวัน
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

สามารถเขียนเป็นโปรแกรมภาษาซีได้ดังนี้

```
#include <stdio.h>
void main()
{
    int number;
    printf("Enter your number : ");
    scanf("%d", &number);
    if (number == 1)
        printf("Monday\n");
    else if (number == 2)
        printf("Tuesday\n");
    else if (number == 3)
        printf("Wednesday\n");
    else if (number == 4)
        printf("Thursday\n");
    else if (number == 5)
        printf("Friday\n");
    else if (number == 6)
        printf("Saturday\n");
    else if (number == 7)
        printf("Sunday\n");
    else
        printf("number is invalid\n");
}
```

1

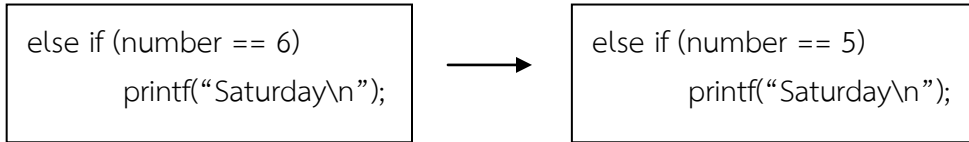
เมื่อสั่งรันโปรแกรมจะปรากฏหน้าจอให้ป้อนตัวเลขดังนี้

Enter your number : _

ถ้าป้อนตัวเลขเท่ากับ 5 ซึ่งทำให้เงื่อนไขของ if ตัวที่ 5 เป็นจริงจะปรากฏข้อความ Friday ออกทางหน้าจอดังนี้

Enter your number : 5
Friday

1 จากโปรแกรม เมื่อพบเงื่อนไขที่เป็นจริง โปรแกรมจะไม่สนใจเงื่อนไขลำดับถัดไปถึงแม้ว่าเงื่อนไขนั้นจะเป็นจริงก็ตาม เช่น เปลี่ยนเงื่อนไขของตัว if ตัวที่ 6 ดังนี้



เมื่อรันโปรแกรม แล้วป้อนตัวเลขเท่ากับ 5 จะปรากฏข้อความ Friday ออกทางหน้าจอ เพราะเป็นเงื่อนไขแรกที่เป็นจริง โดยไม่สนใจเงื่อนไขอื่นที่อยู่ถัดไปทั้งหมด

4. คำสั่ง switch

ในส่วนของฟังก์ชัน switch จะทำการตรวจสอบตัวแปรว่ามีค่าเท่ากับ case ไต ถ้าตรงกับ case ไตก็จะทำงานตามประโยคคำสั่งของ case นั้น การเปรียบเทียบของฟังก์ชัน switch ไม่สามารถเปรียบเทียบค่ามากกว่าน้อยกว่าเหมือนฟังก์ชัน if ได้ และที่สำคัญตัวแปรที่นำมาใช้กับฟังก์ชัน switch จะต้องเป็นข้อมูลชนิดเลขจำนวนเต็มหรือตัวอักษรเท่านั้น ดังนั้น a1, a2 และ a3 อาจจะเป็นค่าคงที่ ตัวอักษร หรือตัวแปรก็ได้ โดยทั่วไปฟังก์ชัน switch นิยมใช้ในการตรวจสอบเงื่อนไข จำนวนหลาย ๆ เงื่อนไขเพราะถ้าใช้ฟังก์ชัน if จะทำให้เกิดความยุ่งยากได้

รูปแบบ switch(ตัวแปร)

```
{
    case a1;
        ประโยคคำสั่ง 1;
        break;
    case a2;
        ประโยคคำสั่ง 2;
        break;
    case a3;
        ประโยคคำสั่ง 3;
        break;
    default;
        ประโยคคำสั่ง ;
}
```

ฟังก์ชัน switch จะทำการตรวจสอบเงื่อนไข case a1 ว่าถูกต้องตรงกับเงื่อนไขหรือไม่ ถ้าตรงตามเงื่อนไขจะทำงานตามประโยคคำสั่งชุดที่ 1 ในกรณีที่ไมตรงกับ case a1 ก็จะทำกรตรวจสอบเงื่อนไข case ต่อไปจนถึงสุดท้าย ถ้าไม่ตรงกับ case ไตเลย โปรแกรมจะไปทำงานตามประโยคคำสั่งที่อยู่หลัง default :

ตัวอย่างโปรแกรมที่ 5

```
#include<stdio.h>
#include<conio.h>
int main( ){
    int day;
    printf("Enter Number of Day : ");
    scanf("%d",&day);
    switch(day) {
        case 1: printf("Monday\n"); break;
        case 2 : printf("Tuesday\n"); break;
        case 3 : printf("Wednesday\n"); break;
        case 4 : printf("Thursday\n"); break;
        case 5: printf("Friday\n"); break;
        case 6 : printf("Saturday\n"); break;
        case 7 : printf("Sunday\n"); break;
        default : printf("*** Not Select ***\n");
    }
    getch();
    return 0;
}
```

1

ผลลัพธ์ คือ

เมื่อสั่งรันโปรแกรม แล้วถ้าป้อนตัวเลขเท่ากับ 5 ซึ่งทำให้เงื่อนไขของ case ตัวที่ 5 เป็นจริงจะปรากฏข้อความ Friday ออกทางหน้าจอดังนี้

```
Enter Number of Day : 5
Friday
```

1

จากโปรแกรม เมื่อพบเงื่อนไขที่เป็นจริง โปรแกรมจะไม่สนใจเงื่อนไขลำดับถัดไป ดังนี้

```
case 5: printf("Friday\n"); break;
```

เมื่อรันโปรแกรม แล้วป้อนตัวเลขเท่ากับ 5 จะปรากฏข้อความ Friday ออกทางหน้าจอ เพราะเป็นเงื่อนไขแรกที่เป็นจริง โดยไม่สนใจเงื่อนไขอื่นที่อยู่ถัดไปทั้งหมด

ที่มา :

ธีรวัฒน์ ประกอบผล. (2552). **คู่มือการเขียนโปรแกรม ภาษา C**. กรุงเทพฯ : ชัคเซส มีเดีย.

สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ. (2553). **หนังสือเรียนรายวิชาเพิ่มเติม เทคโนโลยีสารสนเทศและการสื่อสาร ภาษาซี ชั้นมัธยมศึกษาปีที่ 4-6**. กรุงเทพฯ: โรงพิมพ์ สกสค.