

## ใบความรู้ที่ 10 เรื่อง โครงสร้างควบคุมแบบวนซ้ำ

\*\*\*\*\*

การเขียนโปรแกรมควบคุมทิศทางแบบวนซ้ำ (Loop) หมายถึง การทำคำสั่งซ้ำๆกันหลายครั้งตามเงื่อนไขที่กำหนดไว้ ในการตรวจสอบเงื่อนไข จะมีรูปแบบของการตรวจสอบเงื่อนไขอยู่ 2 แบบ

1. ตรวจสอบก่อนการวนซ้ำ รูปแบบนี้จะตรวจสอบเงื่อนไขว่าเป็นจริงหรือเป็นเท็จ ถ้าเป็นจริงก็จะไปทำงานตามคำสั่งที่กำหนดไว้ และเมื่อทำงานเสร็จก็จะกลับมาทำการตรวจสอบเงื่อนไขอีกครั้ง ทำเช่นนี้ไปเรื่อยๆจนกระทั่งเงื่อนไขได้ค่าเป็นเท็จ ก็จบการวนซ้ำ
2. ตรวจสอบหลังการวนซ้ำ รูปแบบนี้จะทำคำสั่งก่อน ทำคำสั่งเสร็จแล้วจึงค่อยมาตรวจสอบเงื่อนไขว่าเป็นจริงหรือเป็นเท็จ ถ้าเป็นจริงก็จะกลับไปทำคำสั่งเดิมอีกครั้ง ทำเช่นนี้ไปเรื่อยๆ จนกว่าเงื่อนไขที่ได้เป็นเท็จ

**การควบคุมการวนซ้ำ** เพื่อป้องกันการวนซ้ำแบบไม่สิ้นสุด โปรแกรมจำเป็นต้องควบคุมดังต่อไปนี้

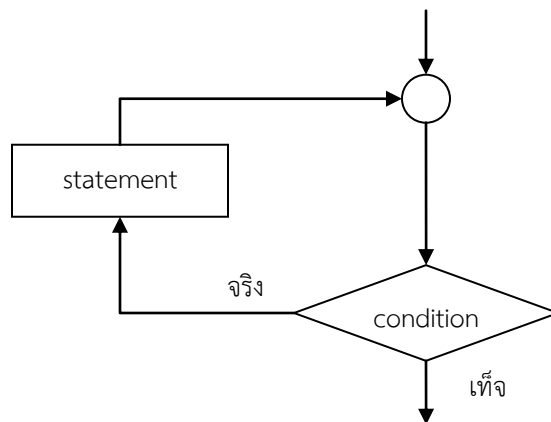
1. การกำหนดตัวแปรควบคุมการวนซ้ำ ก่อนการเริ่มวนซ้ำจะต้องมีการกำหนดตัวแปรขึ้นมาเพื่อใช้ในการตรวจสอบให้สามารถจบการทำงานได้
2. การปรับตัวแปรควบคุมการวนซ้ำ หลังจากทำงานคำสั่งไปแล้ว จำเป็นต้องปรับตัวแปรควบคุมการวนซ้ำทุกครั้ง เพื่อป้องกันไม่ได้วนซ้ำแบบไม่สิ้นสุด

### 1. คำสั่ง while

while เป็นแบบตรวจสอบก่อนการวนซ้ำ มีการตรวจสอบเงื่อนไขก่อนการวนซ้ำ ถ้าเป็นจริงจึงจะเข้าไปทำงานตามคำสั่งที่กำหนดไว้ แต่ถ้าเป็นเท็จก็จะจบการวนซ้ำ มีรูปแบบดังต่อไปนี้

```
while ( condition )  
    statement ;
```

จากรูปแบบข้อความสั่ง while สามารถเขียนเป็นผังงานได้ตามภาพต่อไปนี้



รูปที่ 1 ผังงานข้อความสั่ง while

ข้อสังเกต เช่นเดียวกับคำสั่ง if พบว่า Statement ของ while มีได้เพียงคำสั่งเดียว นั้นหมายความว่าถ้าต้องการเขียนมากกว่า 1 Statement ต้องใช้เครื่องหมาย { }

### ตัวอย่างที่ 1

โปรแกรมแสดงตัวเลขตั้งแต่ 1 ถึง 10

```
#include <stdio.h>
void main()
{
    int number = 1;
    while (number <= 10)
    {
        printf("%d\n", number);
        number++;
    }
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏหน้าจอดังนี้

```
1
2
3
4
5
6
7
8
9
10
```

### ตัวอย่างที่ 2

โปรแกรมรับคะแนนที่ป้อนเข้ามาจากคีย์บอร์ด ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ซึ่งทำให้เงื่อนไขของข้อความสั่ง if เป็นจริง จะแสดงข้อความ Pass ปรากฏออกทางจอภาพ

โปรแกรมสามารถวนซ้ำทำงานไปเรื่อยๆ ได้จนกว่าผู้ใช้ป้อนคะแนนเท่ากับ 0 จึงจะจบการวนซ้ำ

```
#include <stdio.h>
void main()
{
    int score;
    printf("Enter your score : ");
    scanf("%d", &score); ← 1
    while (score != 0)
    {
        if (score >= 50)
            printf("Pass\n");
        else
            printf("Fail\n");
        printf("Enter your score : ");
        scanf("%d", &score); ← 2
    }
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏหน้าจอดังนี้

```
Enter your score : 36
Fail
Enter your score : 60
Pass
Enter your score : 70
Pass
Enter your score : 0
```

1

ข้อความสั่ง while จะตรวจสอบเงื่อนไขก่อนการวนซ้ำ ดังนั้นโปรแกรมจึงจำเป็นต้องเขียนคำสั่งรับค่าคะแนน (score) ก่อนเข้าสู่ข้อความสั่ง while ถ้าเป็นจริง ก็จะทำงาน

2

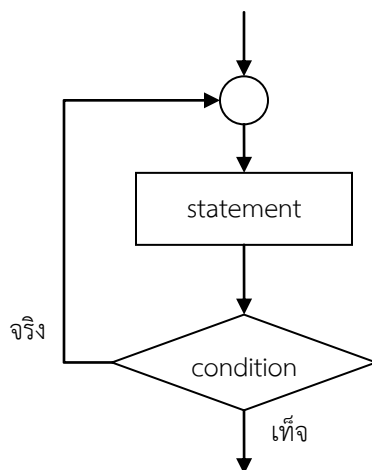
ขั้นตอนสุดท้ายคือการรับค่าคะแนน (score) ก่อนกลับเข้าสู่การตรวจสอบเงื่อนไขของ while อีกครั้ง

## 2. คำสั่ง do-while

do-while เป็นแบบตรวจสอบหลังการวนซ้ำ เมื่อทำงานเสร็จจึงจะตรวจสอบเงื่อนไข ถ้าเป็นจริงจะกลับเข้าไปทำงานตามคำสั่งที่กำหนดไว้ แต่ถ้าเป็นเท็จก็จะจบการวนซ้ำ มีรูปแบบดังต่อไปนี้

```
do
    statement ;
while ( condition )
```

จากรูปแบบข้อความสั่ง do-while สามารถเขียนเป็นผังงานได้ตามภาพต่อไปนี้



รูปที่ 2 ผังงานข้อความสั่ง do while

### ตัวอย่างที่ 1

โปรแกรมแสดงตัวเลขตั้งแต่ 1 ถึง 10

```
#include <stdio.h>
void main()
{
    int number = 1;
    do{
        printf("%d\n", number);
        number++;
    }while (number <= 10);
}
```

โปรแกรมนี้คล้ายกับโปรแกรมที่เขียนด้วยข้อความสั่ง while ต่างกันตรงที่ย้ายเงื่อนไขมาไว้ข้างล่าง และไม่  
ว่าอย่างไรก็ตามโปรแกรมนี้จะแสดงค่าตัวเลขออกทางจอภาพอย่างน้อยที่สุด 1 ค่าเสมอ

## ตัวอย่างที่ 2

โปรแกรมรับคะแนนที่ป้อนเข้ามาจากคีย์บอร์ด ถ้าคะแนนมากกว่าหรือเท่ากับ 50 ซึ่งทำให้เงื่อนไขของ  
ข้อความสั่ง if เป็นจริง จะแสดงข้อความ Pass ปรากฏออกทางจอภาพ

โปรแกรมสามารถวนซ้ำทำงานไปเรื่อยๆ ได้จนกว่าผู้ใช้ป้อนคะแนนเท่ากับ 0 จึงจะจบการวนซ้ำ

```
#include <stdio.h>
void main()
{
    int score;
    do{
        printf("Enter your score : ");
        scanf("%d", &score);
        if (score >= 50)
            printf("Pass\n");
        else
            printf("Fail\n");
    } while (score != 0);
}
```

1 ข้อแตกต่างกับโปรแกรมที่เขียนด้วยข้อความสั่ง while คือ ถึงแม้ว่าจะป้อน 0 เพื่อจบการทำงานแล้วก็  
ตาม โปรแกรมก็ทำงานด้วยการคิดเกรดของค่าที่เท่ากับ 0 ก่อนจึงจะออกจากการวนซ้ำ

## 3. คำสั่ง for

for เป็นแบบตรวจสอบก่อนการวนซ้ำ มีการ  
ตรวจสอบเงื่อนไขก่อนการวนซ้ำ ถ้าเป็นจริงจึงจะเข้าไป  
ทำงานตามคำสั่งที่กำหนดไว้ แต่ถ้าเป็นเท็จก็จะจบการ  
วนซ้ำ มีรูปแบบดังต่อไปนี้

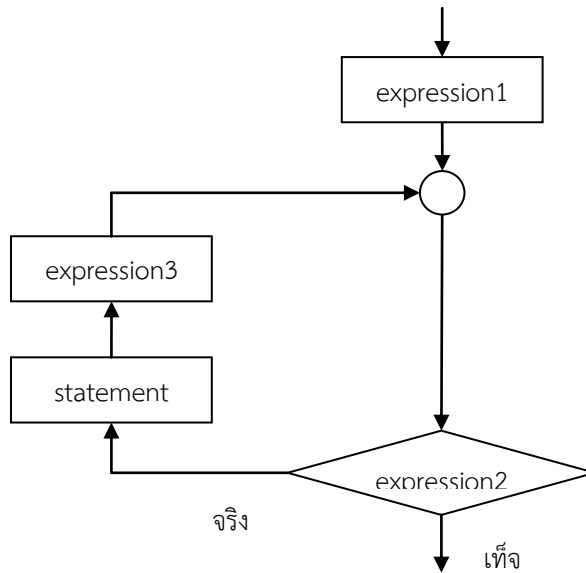
```
for (expression1; expression2; expression3)
    statement ;
```

expression1 หมายถึง การกำหนดตัวแปรควบคุมการวนซ้ำ

expression2 หมายถึง การตรวจสอบเงื่อนไขการวนซ้ำ

expression3 หมายถึง การปรับตัวแปรควบคุมการวนซ้ำ

จากรูปแบบข้อความสั่ง for สามารถเขียนเป็นผังงานได้ตามภาพต่อไปนี้



รูปที่ 1 ผังงานข้อความสั่ง for

### ตัวอย่างที่ 1

โปรแกรมแสดงตัวเลขตั้งแต่ 1 ถึง 10

```
#include <stdio.h>
void main()
{
    for (int number = 1; number <= 4; number++)
    {
        printf("%d\n", number);
    }
}
```

เมื่อสั่งรันโปรแกรมจะปรากฏหน้าจอดังนี้

```
1
2
3
4
```

1

โปรแกรมมีลักษณะคล้ายกับที่เขียนข้อความสั่ง while หรือ do-while ที่เขียนไปก่อนหน้านี้ ส่วนที่แตกต่างก็คือ ตำแหน่งของการเขียน ซึ่งพบว่าได้นำมาเขียนในวงเล็บของ for นั่นเอง และอีกอย่างหนึ่งก็คือ โปรแกรมรู้ว่าจะต้องวนทำซ้ำ 4 รอบ เป็นเหตุให้ควรเขียนโปรแกรมส่วนนี้โดยใช้ข้อความสั่ง for

## ตัวอย่างที่ 2

โปรแกรมรับคะแนนที่ป้อนเข้ามาจากคีย์บอร์ดจำนวนทั้งหมด 15 คะแนน เมื่อรับคะแนนครบ 15 คะแนน ให้คำนวณหาค่าเฉลี่ย

```
#include <stdio.h>
void main()
{
    int score = 0;
    float total = 0, average = 0;
    for (int i = 1; i <= 15 ; i++) ← 1
    {
        printf("Enter your score No. %d : ", i);
        scanf("%d", &score); ← 2
        total = total + score; ← 3
    }
    average = total / 15; ← 4
    printf("Average = %f : ", average);
}
```

- 1 คะแนนมีจำนวนทั้งหมด 15 คะแนนซึ่งเป็นจำนวนคงที่ ดังนั้นจึงเหมาะที่จะใช้ข้อความสั่ง for วนซ้ำรับคะแนนเข้ามา
- 2 รับคะแนนแต่ละรอบแล้ว
- 3 นำคะแนนที่ได้ไปเก็บสะสมไว้ที่ตัวแปรชื่อ total และทั้งหมดนี้เขียนในเครื่องหมาย { } เพื่อระบุว่าเป็น Statement ของ for
- 4 หาค่าเฉลี่ย โดยการนำ total มาหารด้วยค่าคงที่เท่ากับ 15 ผลที่ได้เก็บที่ตัวแปรชื่อ average แล้วจึงแสดงค่าเฉลี่ยออกทางจอภาพ

ที่มา :

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

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