บทที่ 7 กระบวนการเขียนโปรแกรมเชิงโครงสร้าง

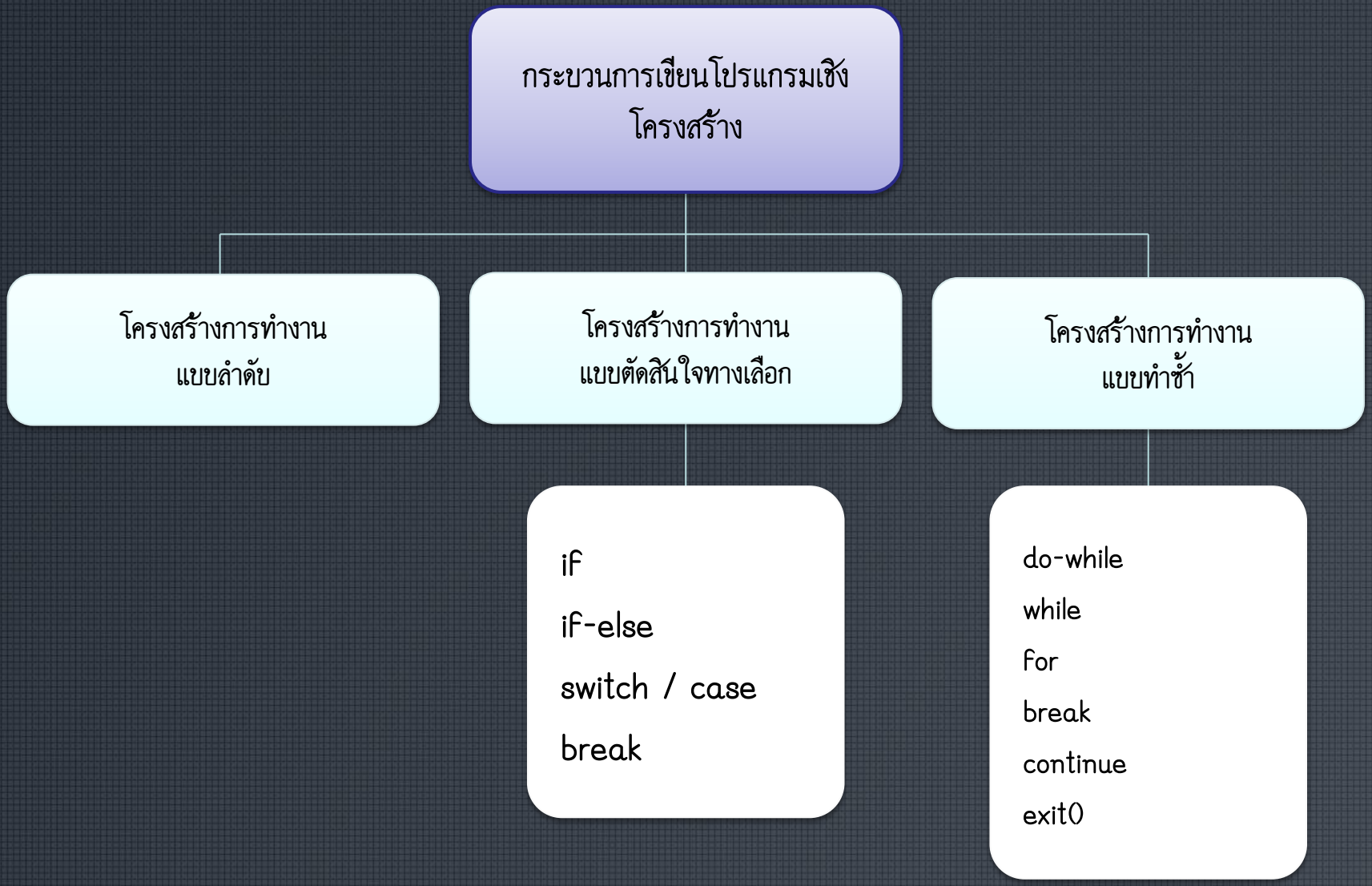
สาระการเรียนรู้

- การเขียนโปรแกรมโครงสร้างการทำงานแบบลำดับ
- การเขียนโปรแกรมโครงสร้างการทำงานแบบตัดสินใจเลือกกระทำตามเงื่อนไข
- การเขียนโปรแกรมโครงสร้างการทำงานแบบทำซ้ำ

สมรรถนะการเรียนรู้

1. เขียนโปรแกรมโครงสร้างการทำงานแบบลำดับได้
2. แปลงสูตรเป็นนิพจน์ในทางคอมพิวเตอร์ได้
3. เขียนโปรแกรมโครงสร้างการทำงานแบบตัดสินใจเลือกกระทำตามเงื่อนไขได้
4. เขียนโปรแกรมโครงสร้างการทำงานแบบทำซ้ำได้
5. ป้อนและทดสอบโปรแกรมตามตัวอย่างได้
6. อธิบายส่วนประกอบในโปรแกรมได้
7. เขียนโปรแกรมตามโจทย์ที่กำหนดให้ได้

แผนผังความคิด (Mind Mapping) ของหน่วยการเรียนรู้



การเขียนโปรแกรมโครงสร้างการทำงานแบบลำดับ

โครงสร้างโปรแกรมแบบลำดับ (Sequence Structure) เป็นโครงสร้างพื้นฐานมีลำดับการทำงานจากบนลงล่าง คือเรียงตามลำดับตั้งแต่ต้นจนจบ เป็นโครงสร้างอย่างง่าย บางทีเรียกโครงสร้างแบบนี้ว่าแบบทางเดียว (Single Pass)

ส่วนประกอบหลักของโปรแกรม ไม่ว่าจะเป็นโครงสร้างแบบใด จะต้องมีส่วนประกอบ 4 ส่วน คือ

1. ส่วนประกาศตัวแปร (Declaration Variables)
2. ส่วนอินพุต (Input)
3. ส่วนประมวลผล (Process)
4. ส่วนเอาต์พุต (Output)

วิธีการประมวลผลตามแบบโครงสร้างแบบลำดับ

จะใช้การคำนวณโดยใช้สูตร (Formula Method)

เช่น

การแปลงผันหน่วย มาตราสำหรับความยาว พื้นที่ หรือ น้ำหนัก การแปลงหน่วยอุณหภูมิ

- สูตรทางพีชคณิต ผลรวมของเลขยกกำลัง ตัวประกอบและการกระจาย
- สูตรการหาความยาว พื้นที่ และปริมาตร

ตัวอย่าง การแปลงผันหน่วยความยาว

1 เมตร = 1.093613 หลา

นำมาเขียนนิพจน์ได้ดังนี้ yards = 1.093613 * meters

1 เมตร = 3.280840 ฟุต

นำมาเขียนนิพจน์ได้ดังนี้ feets = 3.280840 * meters

1 เมตร = 39.37008 นิ้ว

นำมาเขียนนิพจน์ได้ดังนี้ inchs = 39.37008 * meters

ตัวอย่าง การคำนวณตามสูตรทางพีชคณิต

สูตรผลบวกของเลขกำลังของจำนวนเต็มบวก n จำนวน

- $\sum_{k=1}^{n} k = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$
- $\sum_{k=1}^{n} k^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$
- $\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}$
- $\sum_{k=1}^{n} k^4 = \frac{n}{30}(n+1)(2n+1)(3n^2+3n-1)$
- $\sum_{k=1}^{n} k^5 = \frac{n^2}{12}(n+1)^2(2n^2+2n-1)$
- $\sum_{k=1}^{n} k^6 = \frac{n}{42}(n+1)(2n+1)(3n^4+6n^3-3n+1)$
- $\sum_{k=1}^{n} k^7 = \frac{n^2}{24}(n+1)^2(3n^4+6n^3-n^2-4n+2)$

จากสูตรทางพีชคณิตสามารถนำมาเขียนนิพจน์ทางคอมพิวเตอร์ได้ดังนี้

- k1 = n*(n+1)/2
- k2 = n*(n+1)*(2*n+1)/6
- k3 = n*n*(n+1)*(n+1)/4
- k4 = n/30*(n+1)*(2*n+1)*(3*n*n+3*n-1)
- k5 = n*n/12*(n+1)*(n+1)*(2*n*n+2*n-1)
- k6 = n/42*(n+1)*(2*n+1)*(3*n*n*n*n+6*n*n*n-3*n+1)
- k7 = n*n/24*(n+1)*(n+1)*(3*n*n*n*n+6*n*n*n-n*n-4*n+2)

สูตรตัวประกอบและการกระจาย

- $f1 = (a+b)^2 = a^2 + 2ab + b^2$
- $f2 = (a-b)^2 = a^2 - 2ab + b^2$
- $f3 = (a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$
- $f4 = (a-b)^3 = a^3 - 3a^2b + 3ab^2 - b^3$
- $f5 = (a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$
- $f6 = (a-b)^4 = a^4 - 4a^3b + 6a^2b^2 - 4ab^3 + b^4$
- $f7 = a^2 - b^2 = (a-b)(a+b)$
- $f8 = a^3 + b^3 = (a+b)(a^2 - ab + b^2)$
- $f9 = a^3 - b^3 = (a-b)(a^2 + ab + b^2)$
- $f10 = a^4 - b^4 = (a-b)(a^3 + a^2b + ab^2 + b^3)$
- $f11 = a^5 - b^5 = (a+b)(a^4 - a^3b + a^2b^2 - ab^3 + b^4)$

จากสูตรทางพีชคณิตสามารถนำมาเขียนนิพจน์ทางคอมพิวเตอร์ได้ดังนี้

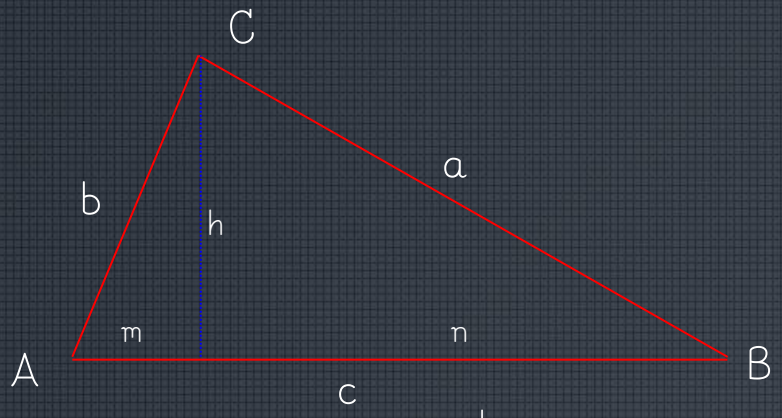
- f1 = a*a + 2*a*b + b*b
- f2 = a*a - 2*a*b + b*b
- f3 = a*a*a + 3*a*a*b + 3*a*b*b + b*b*b
- f4 = a*a*a - 3*a*a*b + 3*a*b*b - b*b*b
- f5 = a*a*a*a + 4*a*a*a*b + 6*a*a*b*b + 4*a*b*b*b + b*b*b*b
- f6 = a*a*a*a - 4*a*a*a*b + 6*a*a*b*b - 4*a*b*b*b + b*b*b*b
- f7 = (a-b) * (a+b)
- f8 = (a-b) * (a*a + a*b + b*b)
- f9 = (a+b) * (a*a - a*b + b*b)
- f10 = (a-b) * (a*a*a + a*a*b + a*b*b + b*b*b)
- f11 = (a+b) * (a*a*a*a - a*a*a*b + a*a*b*b -a*b*b*b + b*b*b*b)

สูตรการหาความยาว พื้นที่ และปริมาตร

สามเหลี่ยม (Triangle)

กำหนดให้ K = พื้นที่, r = รัศมีวงกลมแนบใน, R = รัศมีวงกลมล้อมรอบ

สามเหลี่ยมมุมฉาก



รูปที่ 7.1 รูปสามเหลี่ยมมุมฉาก

มีสูตรที่เกี่ยวข้อง ดังนี้

- มุม — A + B = C = 90 องศา
- พื้นที่ — $K=12ab$
- รัศมีแนบใน — $r=aba+b+c$
- ด้าน — c2 = a2+ b2

$$m = \frac{b^2}{c} \qquad n = \frac{a^2}{c}$$

รัศมีวงกลมล้อมรอบ $R=0.5c$

ความยาว $h=abc$

ตัวอย่างโจทย์

จากรูปสามเหลี่ยมมุมฉาก ถ้าทราบค่า *a* และ *b* จงหาค่าพื้นที่ รัศมีแนบใน รัศมีวงกลมล้อมรอบ ความยาวของ *c, h, m* และ *n*

- อินพุต คือตัวแปร *a* และ *b* เป็นเลขทศนิยมชนิด *double*
- เอาต์พุต คือตัวแปร *c, K, r, R, h, m, n* เป็นเลขทศนิยมชนิด *double*
- นิพจน์

```
c = sqrt(a*a + b*b)
K = 0.5 * a * b
r = (a*b)/(a+b+c)
R = 0.5*c
h = a*b/c
m = b*b/c
n = a*a/c
```

การเขียนโปรแกรมโครงสร้างการทำงาน

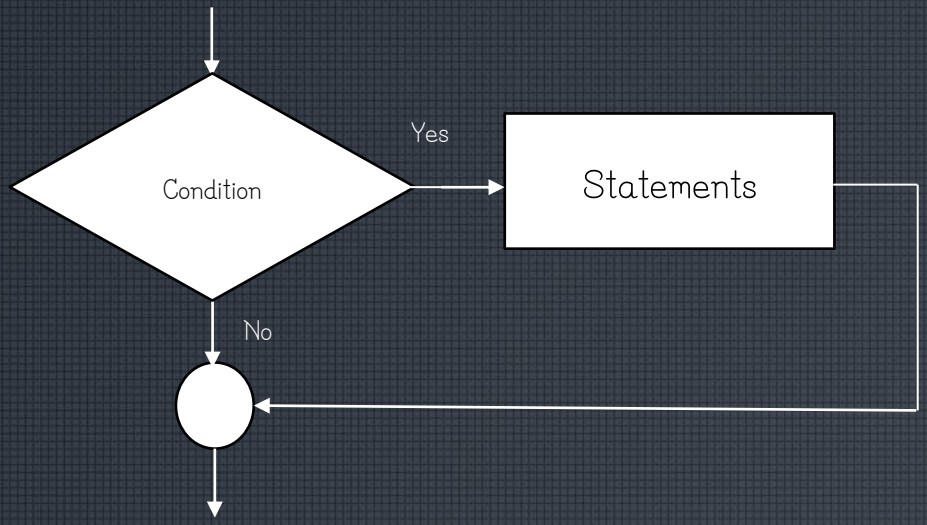
แบบตัดสินใจทางเลือกตามเงื่อนไข

โครงสร้างการทำงานแบบตัดสินใจทางเลือกตามเงื่อนไข หรือแบบเงื่อนไขกรณี หรือแบบการเลือกกระทำตามเงื่อนไข เป็นโครงสร้างที่ให้โปรแกรมสามารถตัดสินใจ (Decision) เลือกทางเลือกตามเงื่อนไข ซึ่งมี 2 ลักษณะใหญ่ ๆ คือ

1. การตัดสินใจเลือกทำกับไม่ต้องทำ หรือ เลือกทำกับข้าม
2. การตัดสินใจเลือกทำอย่างใดอย่างหนึ่ง

การตัดสินใจเลือกทำกับไม่ต้องทำ หรือ เลือกทำกับข้าม

เมื่อเงื่อนไขเป็นจริงจะทำงานในส่วนของกลุ่มประโยคคำสั่ง (Statements) หากเป็นเท็จจะข้ามกลุ่มประโยคคำสั่ง ไปทำงานในคำสั่งถัดไป คือ ข้ามกลุ่มประโยคคำสั่งที่อยู่ในขอบเขต { }

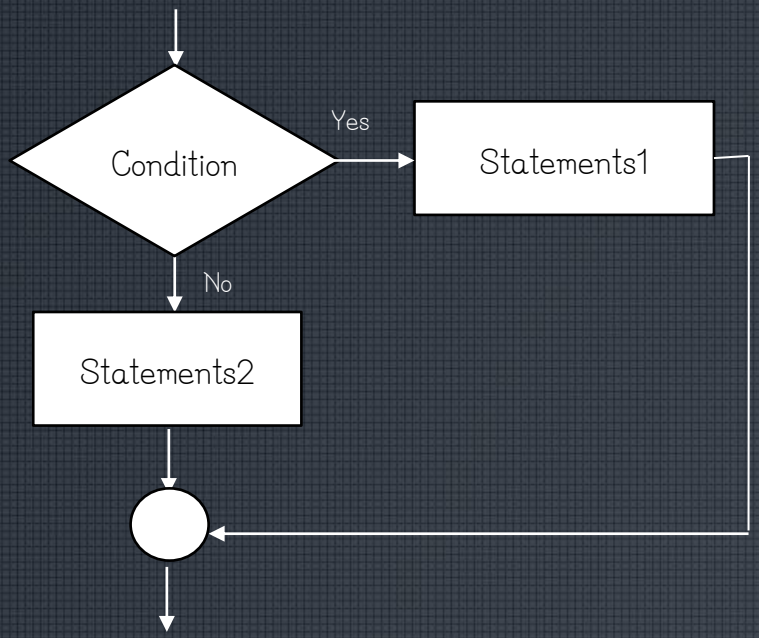


รูปแบบคำสั่ง

```
if  (condition) {
        statements;
}
```

การตัดสินใจเลือกทำอย่างใดอย่างหนึ่ง

เมื่อเงื่อนไขเป็นจริงจะเลือกทำงานในส่วนของกลุ่มประโยคคำสั่งที่ 1 (Statements 1) หากเป็นเท็จจะเลือกทำงานในส่วนของกลุ่มประโยคคำสั่งที่ 2 (Statements 2) คือ เลือกทำอย่างใดอย่างหนึ่ง



รูปแบบคำสั่ง

```
if  (condition) {
        statements 1;
}
else {
        statements 2;
}
```

คำสั่งควบคุมการทำงานแบบทางเลือก

คำสั่ง if

เป็นคำสั่งตัดสินใจเลือกทำ หรือ ไม่ทำ (ข้าม)

รูปแบบ 1

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

รูปแบบ 2

```
if (condition) {
        statement1;
        statement2;
        ...
        statementN;
}
```

คำสั่ง if - else

เป็นคำสั่งตัดสินใจเลือกทำอย่างใดอย่างหนึ่ง

รูปแบบ

รูปแบบที่ 1 ใช้ในกรณีที่ประโยคคำสั่งที่จะเลือกทำมีเพียงประโยคคำสั่งเดียว เพื่อเลือกทำคำสั่งใดคำสั่งหนึ่ง

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

ถ้าเงื่อนไขเป็นจริงจะเลือกทำคำสั่ง statement1

ถ้าเงื่อนไขเป็นเท็จจะเลือกทำคำสั่ง statement2

คำสั่ง if - else

เป็นคำสั่งตัดสินใจเลือกทำอย่างใดอย่างหนึ่ง

- รูปแบบที่ 2 ใช้ในกรณีที่ประโยคคำสั่งที่จะเลือกทำมีตั้งแต่ 1 คำสั่งขึ้นไป เพื่อเลือกทำกลุ่มคำสั่งใดกลุ่มคำสั่งหนึ่ง

```
if (condition) {
        statements1;
}
else {
        statements2;
}
```

ถ้าเงื่อนไขเป็นจริงจะเลือกทำกลุ่มคำสั่ง statements1

ถ้าเงื่อนไขเป็นเท็จจะเลือกทำกลุ่มคำสั่ง statements2

คำสั่ง if ซ้อน if

ใช้ในกรณีที่มีหลายทางเลือก (มากกว่า 2 ทาง)

```
if (condition1) {
    statements_1;
}
else {
    if (condition2) {
                statements_2;
    }
    else {
                if (condition_N) {
                    statements_N;
        }
        else {
                statements-else;
        }
    }
}
```

ถ้าเงื่อนไขที่ 1 (Condition1) เป็นจริงจะเลือกทำกลุ่มคำสั่ง statements_1

ถ้าเงื่อนไขที่ 2 (Condition2) เป็นจริงจะเลือกทำกลุ่มคำสั่ง statements_2

...

ถ้าเงื่อนไขที่ N (Condition_N) เป็นจริงจะเลือกทำกลุ่มคำสั่ง statements_N

แต่ถ้าไม่ตรงกับเงื่อนไขใด ๆ เลย จะเลือกทำกลุ่มคำสั่ง statements-else

ตัวอย่างโปรแกรมตัดเกรด

```
#include <conio.h>
#include <stdio.h>
void main()
{
        int  score, grade;
        printf("Grade Calculate\n\n");
        printf("Enter n : "); scanf("%d",&score);
        if (score >= 80 && score <= 100) grade = 4;
        else if (score >= 70 && score <= 79) grade = 3;
        else if (score >= 60 && score <= 69) grade = 2;
        else if (score >= 50 && score <= 59) grade = 1;
        else grade = 0;
        printf("\nGrade = %d\n",grade);
        getch();
}
```

คำสั่ง switch - case

เป็นคำสั่งตัดสินใจเลือกทำที่มีหลายทางเลือก

รูปแบบ

```
switch (var) {
   case value1 : statement1; break;
   case value2 : statement2; break;
   // any case ...
   case valueN : statementN; break;
   default : statement-default;
}
statement-out_switch
```

ถ้าค่าตัวแปร var ตรงกับ value1 จะทำประโยคคำสั่ง statement1

ถ้าค่าตัวแปร var ตรงกับ value2 จะทำประโยคคำสั่ง statement2

...

ถ้าค่าตัวแปร var ตรงกับ valueN จะทำประโยคคำสั่ง statementN

แต่ถ้าไม่ตรงกับ case ใด ๆ เลย จะทำประโยคคำสั่ง statement-default

คำสั่งควบคุมอื่น ๆ

คำสั่ง break

เป็นคำสั่งที่ต้องใส่ไว้เป็นคำสั่งท้ายสุดของแต่ละ case เพื่อเป็นคำสั่งออกจากคำสั่ง switch ในที่นี้จะไปทำคำสั่งที่อยู่นอกขอบเขตของคำสั่ง switch (ในที่นี้คือ statement-out_switch)

รูปแบบ break;

คำสั่ง continue

เป็นคำสั่งที่ใช้ร่วมกับคำสั่งแบบทำซ้ำ เพื่อสั่งให้โปรแกรมหยุดการทำงานในรอบปัจจุบัน แล้วกลับไปเริ่มทำงานในรอบต่อไป

การเขียนโปรแกรมโครงสร้างการทำงานแบบมีการทำซ้ำ (Repeation)

1) แบบทำก่อนคิด Post-test

คำสั่ง do while

ทำในลูปก่อนแล้วจึงตรวจสอบ

ถ้าเงื่อนไขเป็นจริงจะวนรอบ

ถ้าเงื่อนไขเป็นเท็จจะสิ้นสุดการวนรอบ

ตราบใดที่เงื่อนไขเป็นจริงจะวนรอบ

รูปแบบ

```
do {
        statements;
}
while (condition);
```

การเขียนโปรแกรมโครงสร้างการทำงานแบบมีการทำซ้ำ (Repeation)

2) คิดก่อนทำหรือ แบบตรวจสอบเงื่อนไขก่อน (Pre-test)

คำสั่ง while

ตรวจสอบเงื่อนไขก่อน

ถ้าเงื่อนไขเป็นจริงจะวนรอบ (Loop)

ถ้าเงื่อนไขเป็นเท็จจะสิ้นสุดการวนรอบ

ตราบใดที่เงื่อนไขเป็นจริงจะวนรอบ

รูปแบบ

```
while (condition) {
        statements;
}
```

การเขียนโปรแกรมโครงสร้างการทำงานแบบมีการทำซ้ำ (Repeation)

คำสั่ง For

การทำงานเหมือนกับคำสั่ง while นิยมนำมาเขียนโปรแกรมในการทำงานแบบวนรอบซ้ำ เพราะ ได้นำองค์ประกอบในส่วนที่ 1, 2 และ 3 มาไว้ในคำสั่ง for จึงทำให้การเขียนโปรแกรมผิดพลาดน้อยกว่า

รูปแบบ

```
for (exp1; exp2; exp3) {
        statements;
}
```

โปรแกรมการแสดงเลข 1 ถึง 10 ออกทางจอภาพ

```
#include <conio.h>
#include <stdio.h>
 void main()
{
        int  i;
        printf("For-Loop Programming\n\n");
        for (i = 1; i <= 10; i++)
        {
            printf("%d\n",i);
        }
        getch();
}
```

โปรแกรมการแสดงเลข 10 ถึง 1 ออกทางจอภาพ

```
#include <conio.h>
#include <stdio.h>
 void main()
{
        int  i;
        printf("For-Loop Programming\n\n");
        for (i = 10; i >= 1; i--)
        {
            printf("%d\n",i);
        }
        getch();
}
```

โปรแกรมสูตรคูณ โดยรับแม่สูตรคูณจากแป้นพิมพ์ โดยใช้คำสั่ง for

```
#include <conio.h>
#include <stdio.h>
void main0
{
        int  i, m;
        printf("Multiply Program\n"); printf("Enter m = ");
        scanf("%d",&m);
        printf("\n------------\n");
        printf("%6d",m);
        printf("\n------------\n");
        for (i = 1; i <= 12; i++)
            printf("%2d =%3d\n",i,i*m);
        printf("------------\n");
        getch0;
}
```

โปรแกรมสูตรคูณ โดยรับแม่สูตรคูณจากแป้นพิมพ์ โดยใช้คำสั่ง do while

```
#include <conio.h>
#include <stdio.h>
void main()
{
        int  i, m;
        printf("Multiply Program\n"); printf("Enter m = ");
        scanf("%d",&m);
        printf("\n------------\n");
        printf("%6d",m);
        printf("\n------------\n");
        i = 1;
        do {
            printf("%2d =%3d\n",i,i*m); i++;
        } while (i <= 12);
        printf("------------\n");
        getch();
}
```

โปรแกรมสูตรคูณ โดยรับแม่สูตรคูณจากแป้นพิมพ์ โดยใช้คำสั่ง while

```
#include <conio.h>
#include <stdio.h>
void main()
{
        int  i, m;
        printf("Multiply Program\n"); printf("Enter m = ");
        scanf("%d",&m);
        printf("\n------------\n");
        printf("%6d",m);
        printf("\n------------\n");
        i = 1;
        while (i <= 12) {
            printf("%2d =%3d\n",i,i*m); i++;
        };
        printf("------------\n");
        getch();
}
```

โปรแกรมตรวจสอบการกดคีย์ จนแสดงรหัสคีย์ที่กดจนกว่ากดคีย์ Esc เพื่อจบโปรแกรม

```
#include <conio.h>
#include <stdio.h>
void main()
{
        unsigned char ch;
        clrscr();
        printf("Key Code Program C/C++\n\n");
        printf("Press Esc for Exit.\n\n");
        do
        {
                ch = getch();
                printf("%c : %u\n",ch,ch);
        } while (ch != 27); // 27 is code of Esc Key.
        printf("bye... bye...\n");
        getch();
}
```

โปรแกรมตรวจสอบการป้อนค่าระหว่าง 0-100

```
#include <conio.h>
#include <stdio.h>
void main0
{
   char ch;
   int  x;
        do {
                clrscr0; printf("Test Input Program\n\n");
                do {
                        printf("Enter 0 to 100 : ");
                        scanf("%d",&x);
                        if (x < 0 || x > 100) {
                                printf("Range Error!\n"); fflush(stdin);
                        }
                } while (x < 0 || x > 100);
                printf("\nInput Pass...\n\n");
                printf("Press Esc to Exit or anykey to continue...");
                ch = getch0;
   } while (ch != 27);
}
```

จบการนำเสนอ

บทที่ 7
กระบวนการเขียนโปรแกรมเชิงโครงสร้าง