

## บทที่ 4 คำสั่งควบคุม (Control Statement)

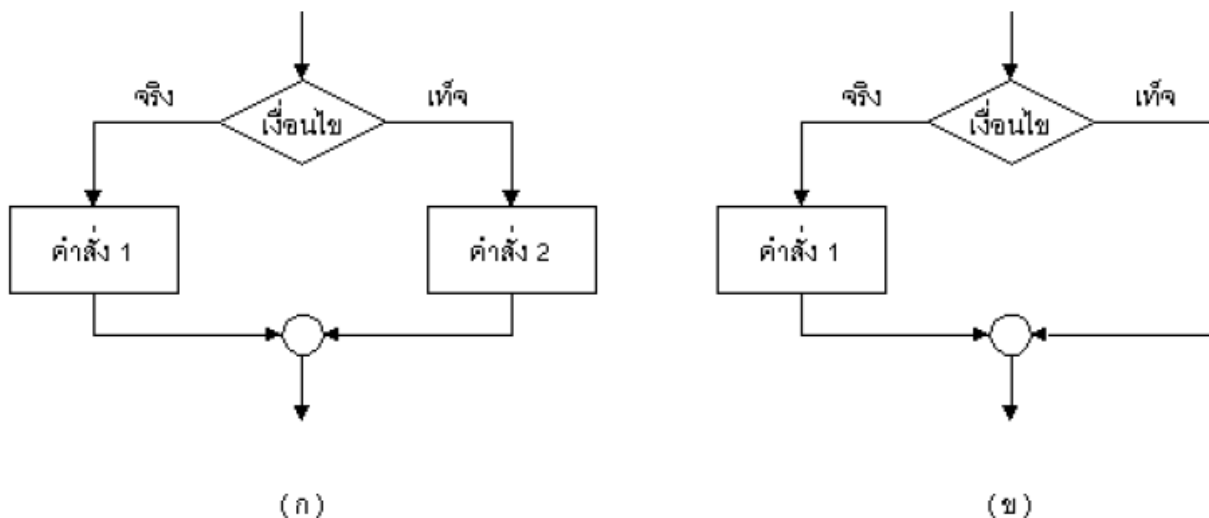
ในการเขียนโปรแกรมแบบโครงสร้าง จะมีรูปแบบการแก้ปัญหาหรือรูปแบบการเขียนโปรแกรมอยู่ 3 ลักษณะ คือ การเขียนแบบลำดับ (Sequential) การเขียนแบบเงื่อนไข (Selection) และการเขียนแบบวนซ้ำ (Repetition) การเขียนทีละคำสั่งจากบนลงจัดเป็นการเขียนแบบลำดับ ส่วนการเขียนแบบเงื่อนไขและการเขียนแบบวนซ้ำนั้นจะต้องใช้คำสั่งควบคุมมาช่วยให้เกิดการเขียนในลักษณะดังกล่าว โดยที่ใช้ภาษาซีมีคำสั่ง if และ switch เพื่อช่วยในการเขียนแบบเงื่อนไข ส่วนในการเขียนแบบวนซ้ำ จะมี 3 คำสั่งคือ for while และ do-while

### 4.1 คำสั่ง IF

คำสั่ง if เป็นคำสั่งที่ใช้ในการเขียนแบบเงื่อนไข ตัวอย่างของประโยคในลักษณะเงื่อนไขเป็นตัวอย่างที่สามารถพบเห็นได้ในชีวิตประจำวันเช่น

- ถ้าวันนี้ฝนไม่ตก ฉันจะเดินไปโรงเรียน แต่ถ้าฝนตก ฉันจะขอให้คุณพ่อไปส่งที่โรงเรียน
- ถ้าฉันสอบได้คะแนนดี คุณพ่อและคุณแม่จะภูมิใจ

จะเห็นว่าประโยคเงื่อนไขดังกล่าวมีอยู่ 2 ลักษณะ คือ ถ้าเงื่อนไขเป็นจริงเกิดเหตุการณ์หนึ่ง แต่ถ้าไม่จริงจะเกิดอีกเหตุการณ์หนึ่ง กับประโยคในลักษณะที่ ถ้าเงื่อนไขเป็นจริงจึงจะเกิดเหตุการณ์ขึ้นเท่านั้น ทั้ง 2 ลักษณะสามารถเขียนเป็นผังงานของงานได้ดังรูปที่ (ก) และ (ข)



แสดงผังงานของประโยคเงื่อนไข

จากผังงานทั้ง 2 จะมีรูปแบบการเขียนคำสั่ง if เกิดขึ้น 2 แบบ

### 4.1.1 คำสั่ง if-else

คำสั่ง if ในรูปแบบแรกจะมีคำสั่งที่ต้องทำทั้งในกรณีที่เงื่อนไขเป็นจริงและเป็นเท็จ โดยใช้พจนานุกรมศาสตร์มาเป็นเครื่องมือช่วยการตรวจสอบเงื่อนไข มีรูปแบบคำสั่ง คือ

```
if ( เงื่อนไข )
    คำสั่งที่ 1;
else
    คำสั่งที่ 2;
```

ตัวอย่างเช่น หากรับข้อมูลจากผู้ใช้และต้องการตรวจสอบว่าเลขที่รับเข้ามามีค่ามากกว่า 10 ให้พิมพ์ข้อความว่า “Number XXXX is over than 10” แต่ถ้า ไม่ใช่ให้พิมพ์ข้อความว่า “Number XXXX is not over than 10” จะเขียนเป็นคำสั่งได้ว่า

```
scanf(“%d”, &number);
if (number > 10)
    printf(“Number %d is over than 10”, number);
else
    printf(“Number %d is not over than 10”, number);
```

แต่หากเงื่อนไขเป็นจริงหรือเท็จแล้ว ต้องทำคำสั่งมากกว่า 1 คำสั่ง จะต้องเขียน if-else ในรูปแบบที่ใช้เครื่องหมาย { } ซึ่งแสดงขอบเขตของการทำเงื่อนไข ครอบคำสั่งที่ต้องทำในแต่ละเงื่อนไข มีรูปแบบดังนี้

```
if ( เงื่อนไข ) {
    คำ สั่งที่ 1;
    คำ สั่งที่ 2;
    ...
} else {
    คำ สั่งที่ 3;
    คำ สั่งที่ 4;
    ...
}
```

ตัวอย่างเหมือนในตัวอย่างก่อนหน้านี้ แต่เพิ่มเงื่อนไขว่า ถ้าเลขนั้นมีค่ามากกว่า 10 ให้ลดเลขนั้นลง 5 แสดงได้ดังตัวอย่าง

```
scanf(“%d”, &number);
if (number > 10) {
    printf(“Number %d is over than 10”, number);
    number = number - 5;
} else
    printf(“Number %d is not over than 10”, number);
```

แต่ถ้าหากมีเงื่อนไขเพิ่มขึ้นอีกว่า ถ้าเลขนั้นไม่มากกว่า 10 ให้เพิ่มค่าเลขนั้นขึ้นอีก 5 สามารถเขียนได้ว่า

```
scanf("%d", &number);
if (number > 10) {
    printf("Number %d is over than 10", number);
    number = number - 5;
} else {
    printf("Number %d is not over than 10", number);
    number = number + 5;
}
```

ทั้งนี้หากมีคำสั่งเพียงคำสั่งเดียวในเงื่อนไข ก็สามารถใช้เครื่องหมาย { } ได้เช่นเดียวกัน เช่น

```
scanf("%d", &number);
if (number > 10) {
    printf("Number %d is over than 10", number);
} else {
    printf("Number %d is not over than 10", number);
}
```

**ตัวอย่างที่ 4.1** โปรแกรมเพื่อตรวจสอบความสูงของนักเรียน 2 คน โดยรับข้อมูลความสูงของนักเรียนทั้งสองมาหาว่าความสูงมากที่สุดคือค่าใด

```
#include <stdio.h>
void main( ) {
    float height1, height2, max;
    printf("Enter first student's height (cm.) : ");
    scanf("%f", &height1);
    printf("Enter second student's height (cm.) : ");
    scanf("%f", &height2);
    if (height1 > height2)
        max = height1;
    else
        max = height2;
    printf("Maximum height is : %.2f cm.", max);
}
```

ผลการทำงานของโปรแกรม

```
Enter first student's height (cm.) : 184.5
Enter second student's height (cm.) : 192.4
Maximum height is : 192.40 cm.
```

**ตัวอย่างที่ 4.2** ให้คำนวณค่าดัชนีมวลกาย (Body Mass Index : BMI) ซึ่งสามารถคิดได้จากสูตร  $BMI = w/h^2$  โดยที่  $w$  แทนน้ำหนักตัวมีหน่วยเป็นกิโลกรัม และ  $h$  แทนความสูงมีหน่วยเป็นเมตร หากค่า BMI อยู่ในช่วง 20-25 ให้ขึ้นข้อความ “Normal BMI.” แต่หากอยู่นอกช่วงดังกล่าวให้ขึ้นข้อความว่า “Dangerous BMI.”

```
#include <stdio.h>
void main( ) {
    float w, h, BMI;
    printf("Enter weight (Kg): ");
    scanf("%f", &w);
    printf("Enter height (M): ");
    scanf("%f", &h);
    BMI = w / (h * h);
    printf("BMI is %.2f", BMI);
    if (BMI >= 20 && BMI <= 25)
        printf("\nNormal BMI.");
    else
        printf("\nDangerous BMI.");
}
```

#### 4.1.2 คำสั่ง if

ในกรณีที่ประโยคเงื่อนไขมีการทำงานเฉพาะเงื่อนไขที่เป็นจริงเท่านั้น โดยไม่มีการทำงานใดในเงื่อนไขที่เป็นเท็จ ดังแสดงในรูปด้านล่าง สามารถเขียนแทนด้วยคำสั่ง if โดยไม่ต้องใส่คำสั่ง else แสดงดังรูปแบบ

```
if ( เงื่อนไข )
    คำสั่งที่ 1;
```

แต่ถ้าเงื่อนไขเป็นจริงแล้วมีการทำคำสั่งมากกว่า 1 คำสั่งขึ้นไป ก็ใช้รูปแบบของเครื่องหมาย { } ซึ่งใช้ในกรณีที่คำสั่งที่ต้องทำในเงื่อนไขการวนซ้ำมากกว่า 1 คำสั่ง เพื่อแสดงขอบเขตของการทำงานนั้น

ตัวอย่าง เช่น ให้รับข้อมูลจำนวนเต็มจากผู้ใช้ หากข้อมูลนั้นมีค่ามากกว่า 60 หรือน้อยกว่า 20 ให้ขึ้นข้อความ “Number XXXX is out of range” เขียนได้ดังตัวอย่าง

```
scanf("%d", &number);
if (number < 20 || number > 60)
    printf("Number %d is out of range", number);
```

**ตัวอย่างที่ 4.3** โปรแกรมสำหรับให้ผู้ใช้คาดเดาตัวอักษรที่มีโปรแกรมได้ตั้งไว้ ถ้าผู้ใช้ป้อนข้อมูลตัวอักษรตรงตามโปรแกรมกำหนดไว้ จะขึ้นคำว่า “Bingo”

```
#include <stdio.h>
#define ANS 'G'
void main() {
    char ch;
    printf("Enter character (a-z/A-Z) : ");
    scanf("%c", &ch);
    if (ch == ANS)
        printf("Bingo");
}
```

**ผลการทำงานของโปรแกรม**  
 Enter character (a-z/A-z) : G  
 Bingo

**ตัวอย่างที่ 4.4** แสดงโปรแกรมเพื่อรับข้อมูลเลขจำนวนเต็ม 2 จำนวนจากผู้ใช้ หากค่าแรกที่รับมามีค่ามากกว่าค่าหลังให้ขึ้นข้อความว่า “First value more than second value”

```
#include <stdio.h>
void main() {
    int a, b;
    printf("Enter A : ");
    scanf("%d", &a);
    printf("Enter B : ");
    scanf("%d", &b);
    if (a > b)
        printf("First value more than second value");
}
```

#### 4.1.3 คำสั่ง if แบบซับซ้อน

ในบางกรณีประโยคเงื่อนไขอาจจะมีควาซับซ้อน มีการเปรียบเทียบเงื่อนไขเดียวกันกับหลายค่า เช่น ให้รับข้อมูลชั้นปีของนักศึกษาและให้พิมพ์ข้อความตรงกับชั้นปี กำหนดว่าชั้นปีที่ 1 พิมพ์ว่า “Freshman” ชั้นปีที่ 2 พิมพ์ว่า “Sophomore” ชั้นปีที่ 3 พิมพ์ว่า “Junior” ชั้นปีที่ 4 พิมพ์ว่า “Senior” ชั้นปีอื่น ๆ พิมพ์ว่า “Super”

```
scanf("%d", &year);
if (year == 1)
    printf("Freshman");
else if (year == 2)
    printf("Sophomore");
else if (year == 3)
    printf("Junior");
else if (year == 4)
    printf("Senior");
else
    printf("Super");
```

นอกจากนี้บางเงื่อนไขอาจมีความซับซ้อน เช่น พิจารณาข้อมูลนักศึกษา ให้ตรวจสอบว่าถ้าเป็นนักศึกษาเพศชายมีความสูงตั้งแต่ 180 ซม. ขึ้นไป ให้ขึ้นข้อความแนะนำว่าควรจะสมัครเข้าชมรมบาสเก็ตบอล แต่ถ้าเป็นเพศหญิงมีความสูงตั้งแต่ 170 ซม. ขึ้นไป ให้ขึ้นข้อความแนะนำว่าควรจะสมัครเข้าชมรมวอลเลย์บอล

```
if (gender == 'M' && height > 180)
    printf("Basketball");
else if (gender == 'F' && height > 170)
    printf("Volleyball");
```

สามารถเขียนในอีกลักษณะหนึ่งได้ว่า

```
if (gender == 'M') {
    if (height > 180)
        printf("Basketball");
} else { /* ไม่ใช่เพศชาย เพราะฉะนั้นเป็นเพศหญิง */
    if (height > 170)
        printf("Volleyball");
}
```

การเขียนในลักษณะดังกล่าว จะต้องระมัดระวังเรื่องการใช้เครื่องหมาย { } ให้ถูกต้อง พิจารณาจากตัวอย่างต่อไปนี้

```
if (gender == 'M')
    if (height > 180)
        printf("Basketball");
else
    if (height > 170)
        printf("Volleyball");
```

เหมือนกัน

```
if (gender == 'M')
    if (height > 180)
        printf("Basketball");
else if (height > 170)
    printf("Volleyball");
```

จะเห็นว่า การเว้นย่อหน้าไม่เกี่ยวข้องกับการจับคู่ของเงื่อนไข if-else เวลาที่คอมไพเลอร์มองดูโปรแกรม ไม่ได้ดูจากการย่อหน้า แต่ดูจากสัญลักษณ์ต่าง ๆ การตีความจึงแตกต่างจากความตั้งใจของผู้เขียนโปรแกรม เพราะฉะนั้นหากไม่แน่ใจ การจับคู่เงื่อนไขใด ให้ใช้เครื่องหมาย { } เป็นตัวบอกขอบเขตการทำงานของคำสั่ง if-else นั้น ๆ โดยที่จำนวนปีกกาเปิดในโปรแกรม จะต้องเท่ากับจำนวนปีกกาปิดในโปรแกรมเสมอ

**ตัวอย่างที่ 4.5** เขียนโปรแกรมเพื่อรับข้อมูลคะแนนสอบของนักศึกษา และให้พิมพ์เกรดที่นักศึกษาได้รับจากเงื่อนไขการให้ลำดับชั้น ดังนี้

คะแนนต่ำกว่า 50 ได้เกรด F

คะแนนต่ำกว่า 60 ได้เกรด D

คะแนนต่ำกว่า 70 ได้เกรด C

คะแนนต่ำกว่า 85 ได้เกรด B

คะแนนตั้งแต่ 85 ขึ้นไปได้เกรด A

```
#include <stdio.h>
void main( ) {
    float score;
    printf("Enter score : ");
    scanf("%f", &score);
    if (score < 50)
        printf("Grade F");
    else if (score < 60)
        printf("Grade D");
    else if (score < 70)
        printf("Grade C");
    else if (score < 85)
        printf("Grade B");
    else
        printf("Grade A");
}
```

**ตัวอย่างที่ 4.6** ให้รับข้อมูลจำนวน 3 จำนวนจากผู้ใช้ และให้หาค่ามากที่สุดมีค่าเท่าใด

```
#include <stdio.h>
void main( ) {
    int first, second, third, max;
    printf("Enter first number : ");
    scanf("%d", &first);
    printf("Enter second number : ");
    scanf("%d", &second);
    printf("Enter third number : ");
    scanf("%d", &third);
    max = first;
```

```

if (max > second)
    max = second;
else if (max > third)
    max = third;
printf("Maximum number is %d", max);
}

```

ในที่นี้ลองพิจารณาตัวอย่างนิพจน์ กับการใช้งานประโยคเงื่อนไข สิ่งที่ต้องระวังคือ หากใช้เครื่องหมาย &&(Logical And) ถ้านิพจน์แรกเป็นเท็จ จะทำให้นิพจน์นั้นเป็นเท็จเสมอ และ ถ้าเป็นเครื่องหมาย || (Logical Or) หากนิพจน์แรกเป็นจริง จะทำให้นิพจน์เป็นจริงเสมอ สิ่งที่ตามมาคือ คอมไพเลอร์จะไม่มีการประมวลผลคำสั่งในนิพจน์ที่คู่กัน

```

int a = 10;
if (a > 5 || a++ > 10)
    a = a * 2;
printf("a = %d", a);

```

จะได้คำตอบว่า a= 20 ทั้งนี้หากสังเกตนิพจน์ที่ 2 ของการเปรียบเทียบจะเห็นว่า ผู้เขียนโปรแกรมตั้งใจให้มีการเพิ่มค่าของ a ขึ้นอีก 1 ก่อนที่จะทำงานใด ๆ แต่เนื่องจากนิพจน์แรกเป็นจริง ทำให้สรุปได้ว่าทั้งนิพจน์นี้เป็นจริง จึงไม่มีการประมวลผลนิพจน์ที่ 2 ค่า a จึงไม่เพิ่มขึ้น ซึ่งอาจจะทำให้ค่าที่ได้ผิดจากความตั้งใจของผู้เขียนโปรแกรม

ตัวอย่างที่ใช้เครื่องหมาย &&

```

int a = 10;
if (a < 5 && a++ > 10)
    a = a * 2;
printf("a = %d", a);

```

จะได้คำตอบคือ a = 10 เนื่องจากนิพจน์แรกเป็นเท็จ ทำให้ไม่มีการประมวลผลนิพจน์ที่ 2 ค่า a จึงไม่มีการเพิ่มขึ้น

นอกจากนี้หากเปรียบเทียบตัวดำเนินการเงื่อนไขที่ได้กล่าวไปแล้ว สามารถเปรียบเทียบกับการทำงานประโยคเงื่อนไข if-else ดังตัวอย่าง

```
x = (y < 0) ? -y : y ;
```

หากเขียนคำสั่งดังกล่าวเป็นประโยคเงื่อนไข if-else จะได้

```

if (y < 0)
    x = -y;
else
    x = y;

```



## 4.2 คำสั่ง switch

คำสั่ง switch เป็นคำสั่งที่ใช้ในการเขียนประโยคเงื่อนไข มักจะใช้กับกรณีที่เป็นเงื่อนไข if แบบซับซ้อน ตัวอย่างเช่นในเรื่องของการตรวจสอบชั้นปีของนักศึกษา และให้พิมพ์ข้อความตามชั้นปีที่กำหนด ดังตัวอย่างที่แสดงในหัวข้อ 4.1.3 จะเห็นว่ามีการตรวจสอบนิพจน์เงื่อนไข คือ ชั้นปีของนักศึกษาในทุกเงื่อนไขเหมือนกัน ประโยคในลักษณะเช่นนี้สามารถใช้คำสั่ง switch มาช่วยในการเขียน เพื่อช่วยให้อ่านเข้าใจได้มากยิ่งขึ้น ทั้งนี้เงื่อนไขที่จะนำมาตรวจสอบในคำสั่ง switch ได้จะต้องมีค่าเป็นเลขจำนวนเต็มหรือตัวอักษรเท่านั้น ไม่สามารถใช้กับการตรวจสอบสตริง หรือข้อมูลที่มีลักษณะเป็นช่วง มีรูปแบบของคำสั่ง switch ดังนี้

```
switch ( เงื่อนไข ) {
    case ค่าคงที่1 : คำ สั่ง1 ;
    case ค่าคงที่2 : คำ สั่ง2 ;
    .....
    .....
    default : คำ สั่ง N ;
}
```

การทำงานของคำสั่ง switch จะตรวจสอบเงื่อนไขว่าตรงกับค่า case ไต ก็จะไปทำงานที่คำสั่งที่อยู่ใน case นั้น คำสั่งหนึ่งที่มีมักจะใช้คู่กับคำสั่ง switch คือ คำสั่ง break คำสั่งนี้ใช้ในการบอกให้โปรแกรมหยุดการทำงาน และกระโดดออกจากขอบเขตของ { } ที่ใกล้ที่สุด ซึ่งสามารถใช้คำสั่งนี้ร่วมกับคำสั่งวนซ้ำอื่น ๆ อีกด้วย พิจารณาการทำงานของคำสั่ง switch จากตัวอย่าง

ตัวอย่าง (ก)	ตัวอย่าง (ข)
<pre>int a = 2; switch (a) {     case 1 : printf("11111\n");     case 2 : printf("22222\n");     case 3 : printf("33333\n");     default : printf("AAAAA\n"); }</pre>	<pre>int a = 2; switch (a) {     case 1 : printf("11111\n");                 break;     case 2 : printf("22222\n");                 break;     case 3 : printf("33333\n");                 break;     default : printf("AAAAA\n"); }</pre>
<p>ผลการทำงาน</p> <pre>22222 33333 AAAAA</pre>	<p>ผลการทำงาน</p> <pre>22222</pre>

จากตัวอย่าง(ก) เมื่อโปรแกรมตรวจสอบนิพจน์ว่าตัวแปร a มีค่าเท่ากับ 2 จะมาทำงานที่คำสั่ง case 2 พิมพ์ค่า 22222 และทำคำสั่งต่อ ๆ มาได้ผลลัพธ์ดังตัวอย่าง

ส่วนตัวอย่าง(ข) มีการใช้คำสั่ง break เมื่อมีการตรวจสอบว่าตัวแปร a มีค่าเท่ากับ 2 จะมาทำงานที่คำสั่ง case 2 พิมพ์ค่า 22222 และทำคำสั่ง break ซึ่งจะทำให้การทำงานกระโดดออกจากขอบเขตของเครื่องหมาย { } ที่ใกล้ที่สุด ได้ผลลัพธ์ดังตัวอย่าง

คำสั่ง default ใน switch จะมีค่าเหมือนกับ else ในคำสั่ง if-else ก็คือค่าใด ๆ ก็ตามที่ไม่ใช่ค่าที่กำหนดใน case จะมาทำที่คำสั่ง default ซึ่งคำสั่ง default นี้จะมีหรือไม่มีก็ได้ หากเขียนคำสั่ง switch แทนคำสั่ง if-else ของการพิมพ์ค่าของชั้นปีนักศึกษา จะได้ดังตัวอย่างด้านล่าง

<pre>scanf("%d", &amp;year); if(year == 1)     printf("Freshman"); else if(year == 2)     printf("Sophomore"); else if(year == 3)     printf("Junior"); else if(year == 4)     printf("Senior"); else     printf("Super");</pre>	<pre>scanf("%d", &amp;year); switch(year) {     case 1 : printf("Freshman");              break;     case 2 : printf("Sophomore");              break;     case 3 : printf("Junior");              break;     case 4 : printf("Senior");              break;     default : printf("Freshman"); }</pre>
--	--

นอกจากนี้ยังเขียนคำสั่ง switch ในลักษณะอื่น ๆ แสดงดังตัวอย่างด้านล่าง

**ตัวอย่างที่ 4.7** เขียนโปรแกรมเพื่อรับข้อมูลตัวอักษรจากผู้ใช้ หากผู้ใช้ป้อนตัวอักษร a, b, x ให้ขึ้นข้อความว่า "Hanaga" ป้อนตัวอักษร u, d, p ให้ขึ้นข้อความว่า "Bingo" ป้อนตัวอักษร g ให้ขึ้นข้อความว่า "Google" ป้อนตัวอักษรอื่น ๆ ให้ขึ้นข้อความว่า "Yappadappadoo"

```
#include <stdio.h>
void main( ) {
    char ch;
    printf("Enter character : ");
    scanf("%c", &ch);
    switch ( ch ) {
        case 'a' :
        case 'b' :
        case 'x' : printf("Hanaga");
                  break;
        case 'u' :
        case 'd' :
        case 'p' : printf("Bingo");
                  break;
```

```

        case 'g' : printf("Google");
                break;
        default : printf("Yappadappadoooo");
    }
}

```

**ตัวอย่างที่ 4.8** เขียนโปรแกรมเพื่อคำนวณค่าจ้างของพนักงาน โดยกำหนดให้รับข้อมูลจำนวนชั่วโมงทำงานและประเภทพนักงาน ซึ่งพนักงานแต่ละประเภทได้รับค่าจ้างต่อชั่วโมงต่างกัน ดังนี้

ประเภทของงาน	อัตราค่าจ้าง/ชั่วโมง
0	30
1	40
2	45

```

#include <stdio.h>
#include <conio.h>
void main() {
    float workRate, totalPay;
    int workHours;
    char type;
    clrscr();
    printf("Enter type (1-3) : ");
    scanf("%c", &type);
    fflush();
    if (type == '1' || type == '2' || type == '3') {
        switch (type) {
            case '1' : workRate = 30.0f;
                       break;
            case '2' : workRate = 40.0f;
                       break;
            default : workRate = 45.0f;
        }
        printf("Enter work hours : ");
        scanf("%d", &workHours);
        fflush();
        totalPay = workHours * workRate;
        printf("Rate is %.2f, total pay = %.2f", workRate, totalPay);
    } else
        printf("!!!Error : incorrect employee type");
    getch();
}

```