

บทที่ 4-1

การควบคุมทิศทางการทำงานของโปรแกรมแบบเลือกทำ

Control Structure (Selection)

การควบคุมทิศทางการทำงานของโปรแกรม

การควบคุมการทำงานของโปรแกรมประกอบด้วย 3 รูปแบบ

- การควบคุมทิศทางการทำงานแบบลำดับ (Sequence)
- การควบคุมทิศทางการทำงานแบบเลือกทำ (Selection)
- การควบคุมทิศทางการทำงานแบบทำซ้ำ (Repetition)

การควบคุมแบบเลือกทำ

การควบคุมทิศทางการทำงานแบบเลือกทำ (Selection)

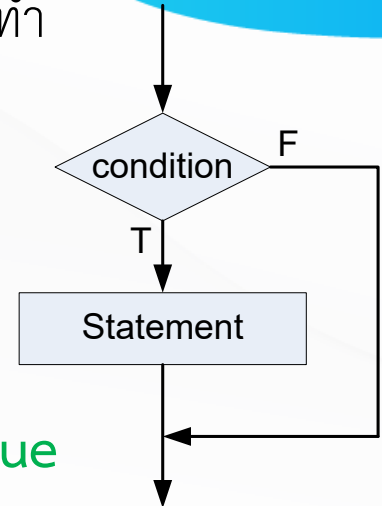
- คำสั่ง if
- คำสั่ง if-else
- คำสั่ง if-else if
- คำสั่ง switch

การควบคุมทิศทางแบบเลือกทำ if

- **if** ใช้ในกรณีที่มีทางเลือกอยู่เพียงทางเลือกเดียวคือทำหรือไม่ทำ

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

- **condition** : เงื่อนไขที่ให้ผลลัพธ์เป็น **true** หรือ **false**
- **statement** : คำสั่งที่จะให้ทำหากผลลัพธ์ของเงื่อนไขเป็น **true**
- หากมีคำสั่งมากกว่าหนึ่งคำสั่งจะต้องมีเครื่องหมายปีกกาครอบคำสั่งไว้



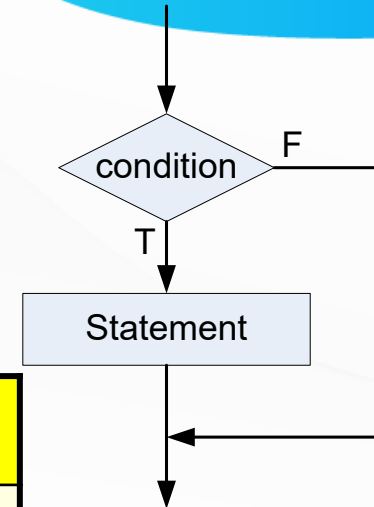
```
if(condition)
{
    statement_1;
    statement_2;
    ...
    statement_n;
}
```

การควบคุมทิศทางแบบเลือกทำ if

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

- **condition** : เงื่อนไขที่ให้ผลลัพธ์เป็น **true** หรือ **false**
 - ใช้ตัวดำเนินการเปรียบเทียบ (Relational Operators)
- ทบทวนตัวดำเนินการเปรียบเทียบ

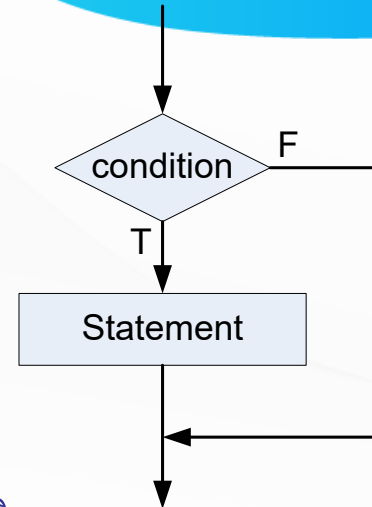
operator	ความหมาย	ตัวอย่าง	ผลลัพธ์
<	น้อยกว่า	5 < 4	0
<=	น้อยกว่าหรือเท่ากับ	5 <= 5	1
==	เท่ากับ	0 == 0	1
>	มากกว่า	5 > 4	1
>=	มากกว่าหรือเท่ากับ	5 >= 4	1
!=	ไม่เท่ากับ	0 != 0	0



การควบคุมทิศทางแบบเลือกทำ if

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

- condition : เงื่อนไขที่ให้ผลลัพธ์เป็น **true** หรือ **false**
 - ใช้ตัวดำเนินการเปรียบเทียบ (Relational Operators)
<, <=, ==, >, >= และ != ให้ผลคือ **0** หรือ **1**
- ทบทวนตัวดำเนินการเปรียบเทียบ
 - ข้อมูลตรรกะ (Boolean Data Type) จะเป็นค่าทางลอจิก
 - ค่าเท็จ (**False**) แทนค่าด้วยเลข **0**
 - ค่าจริง (**True**) แทนค่าด้วยเลข **1**

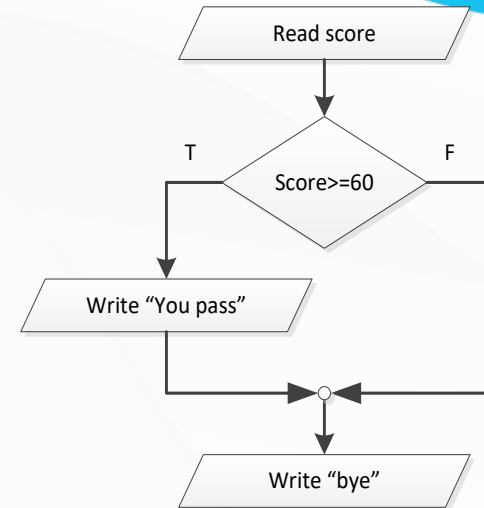


หมายเหตุ นอกเหนือจากผลลัพธ์ของ ตัวดำเนินการ
เปรียบเทียบ **ค่าจริง** คือ **ค่าที่ไม่เท่ากับ 0**

เช่น -1 4 25 356

ตัวอย่างการใช้งาน if

```
1 scanf ("%d" , &score) ;  
2 if (score >= 60)  
3     printf("You pass\n") ;  
4 printf("bye") ;
```



score = 60

ผลลัพธ์ที่ได้ ?

You pass

bye

score = 59

ผลลัพธ์ที่ได้ ?

bye

score = 80

ผลลัพธ์ที่ได้ ?

You pass

bye

ตัวอย่างการใช้งาน if

```
1  #include <stdio.h>
2  int x,y;
3  int main(void)
4  {
5      printf("Enter total score : ");
6      scanf("%d",&x);
7
8      printf("Enter number of students : ");
9      scanf("%d",&y);
10
11     if(y==0)
12         printf("Divided by zero !\n");
13 }
```

```
Enter total score : 500
Enter number of students : 0
Divided by zero !
```


ตัวอย่างการใช้งาน if

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int magic = 123;    // กำหนดค่า 123 ให้กับตัวแปร
5      int guess;
6      printf("Enter your guess: ");
7      scanf("%d", &guess);
8
9      if (guess == magic)
10         printf("**Right**");
11 }
```

ผลลัพธ์ที่ได้ ?

Enter your guess: 100
← ไม่แสดงค่าอะไรเลย

ผลลัพธ์ที่ได้ ?

Enter your guess: 123
Right

ตัวอย่างการใช้งาน if

จงรับค่าจำนวนเต็มเข้ามา ไม่ว่าจะป็นเลขบวกหรือลบ จากนั้นจะแสดงข้อมูลเป็นเลขบวกทางหน้าจอ โดยข้อมูลที่รับมานั้นจะถูกตรวจสอบว่าเป็นเลขลบหรือไม่ถ้าเป็นให้คูณด้วย -1

```
1  int main(void)
2  {
3      int x;
4      printf("INPUT NUMBER: ");
5      scanf("%d", &x);
6
7      if (x<0)
8          x=x*-1;
9
10     printf("OUTPUT %d", x);
11 }
```

ผลลัพธ์ที่ได้ ?

INPUT NUMBER: 10
OUTPUT 10

INPUT NUMBER: -5
OUTPUT 5

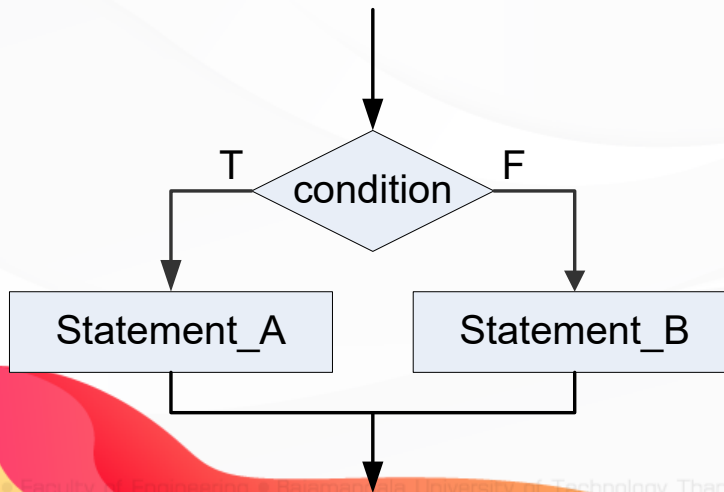
การควบคุมทิศทางแบบเลือกทำ if-else

if-else ใช้ในกรณีที่ มีทางเลือกอยู่สองทาง

หาก **condition** เป็น **true** จะทำ statement ชุด A

หาก **condition** เป็น **false** จะทำ statement ชุด B

```
if(condition)
    statement_A;
else
    statement_B;
```



```
if(condition)
{
    statement_A1;
    ...
    statement_An;
}
else
{
    statement_B1;
    ...
    statement_Bn;
}
```

ตัวอย่างการใช้งาน if-else (1)

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int score;
5      scanf("%d", &score);
6
7      if (score >= 60)
8          printf("You pass ");
9      else
10         printf("You fail ");
11
12         printf("Have a nice day");
13 }
```

ผลลัพธ์ที่ได้ ?

59

You fail Have a nice day

ผลลัพธ์ที่ได้ ?

79

You pass Have a nice day

ตัวอย่างการใช้งาน if-else (2)

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int month;
5      scanf("%d", &month);
6
7      if (month >= 6 && month <= 9)
8          printf("Semester : 1 ");
9      else
10         printf("See you next year!");
11     return(0);
12 }
```

ตัวดำเนินการลอจิก
(Logical Operators)

ผลลัพธ์ที่ได้ ?

9
Semester : 1

ผลลัพธ์ที่ได้ ?

4
See you next year!

ทบทวนตัวดำเนินการเปรียบเทียบ

- condition : เงื่อนไขที่ให้ผลลัพธ์เป็น **true** หรือ **false**
- ใช้ตัวดำเนินการเปรียบเทียบ (Relational Operators)

<, <=, ==, >, >= และ != ให้ผลลัพธ์คือ 0 หรือ 1

- ตัวดำเนินการลอจิก (Logical Operators) &&, || และ ! (NOT)

operator	ความหมาย	ตัวอย่าง	ผลลัพธ์
&&	AND	1 && -1	1
	OR	1 0	1
!	NOT	!4	0

สรุป

AND (&&) ผลลัพธ์เป็น **true** ถ้าค่าทั้งสองค่าเป็น **true**

OR (||) ผลลัพธ์เป็น **false** ถ้าค่าทั้งสองค่าเป็น **false**

NOT (!) เปลี่ยนค่าจาก **true** เป็น **false** จาก **false** เป็น **true**

ตัวอย่างการใช้งาน if-else (2)

```
1  #include <stdio.h>
2  int choice;
3  float radius, circum, area;
4  int main(void)
5  {
6      printf("1.Cirumference of the circle\n");
7      printf("2.Area of the circle\n");
8      printf("Enter your choice 1 or 2 : ");
9      scanf("%d",&choice);
10
11     printf("Enter radius of the circle : ");
12     scanf("%f",&radius);
13
14     if(choice == 1) {
15         circum = 2*3.14156*radius;
16         printf("Circumference of the circle =%f\n",circum);
17     }
18     else {
19         area = 3.14156*radius*radius;
20         printf("Area of the circle = %f\n",area);
21     }
22 }
```

ผลลัพธ์ที่ได้ ?

1.Cirumference of the circle
2.Area of the circle

Enter your choice 1 or 2 : 1
คำนวณอะไร ?

Enter your choice 1 or 2 : 2
คำนวณอะไร ?

Enter your choice 1 or 2 : 3
คำนวณอะไร ?

ตัวอย่างการใช้งาน if-else (3)

จงรับค่าเลขจำนวนเต็มจากแป้นพิมพ์ จากนั้นจะนำไปเปรียบเทียบกับ 0
ว่าค่าที่รับเข้ามานั้นเป็นบวกหรือลบ จากนั้นจะแสดงประเภทของตัวเลขออกมา

```
1  int main(void)
2  {
3      int num;
4      printf("enter a number: ");
5      scanf("%d", &num);
6
7      if (num < 0)    //ตรวจสอบว่าค่าที่รับน้อยกว่า 0 หรือไม่
8          printf("number is negative"); //ถ้าน้อยกว่า 0 บอกว่าเป็นค่าลบ
9      else
10         printf("number is positive"); //ถ้าค่ามากกว่า 0 บอกว่าเป็นค่าบวก
11
12 }
```


ตัวอย่างการใช้งาน if-else (4)

โปรแกรมต่อไปนี้เป็นโปรแกรมคำนวณราคาต้นทุนสินค้า ถ้าหากผลิตมากกว่า 10 ชิ้นจะขึ้นละ 6.5 บาท แต่ถ้าไม่เกิน 10 ชิ้นจะราคาขึ้นละ 7 บาท

```
1  int main(void)
2  {
3      int Number;
4      float cost;
5      printf("Enter number : ");
6      scanf("%d", &Number);
7
8      if(Number > 10)
9          cost = Number * 6.5;
10     else
11         cost = Number * 7;
12     printf("Cost = %0.2f\n", cost);
13 }
```

ผลลัพธ์ที่ได้ ?

Enter number : 5
Cost = 35.00

Enter number : 11
Cost = 71.50

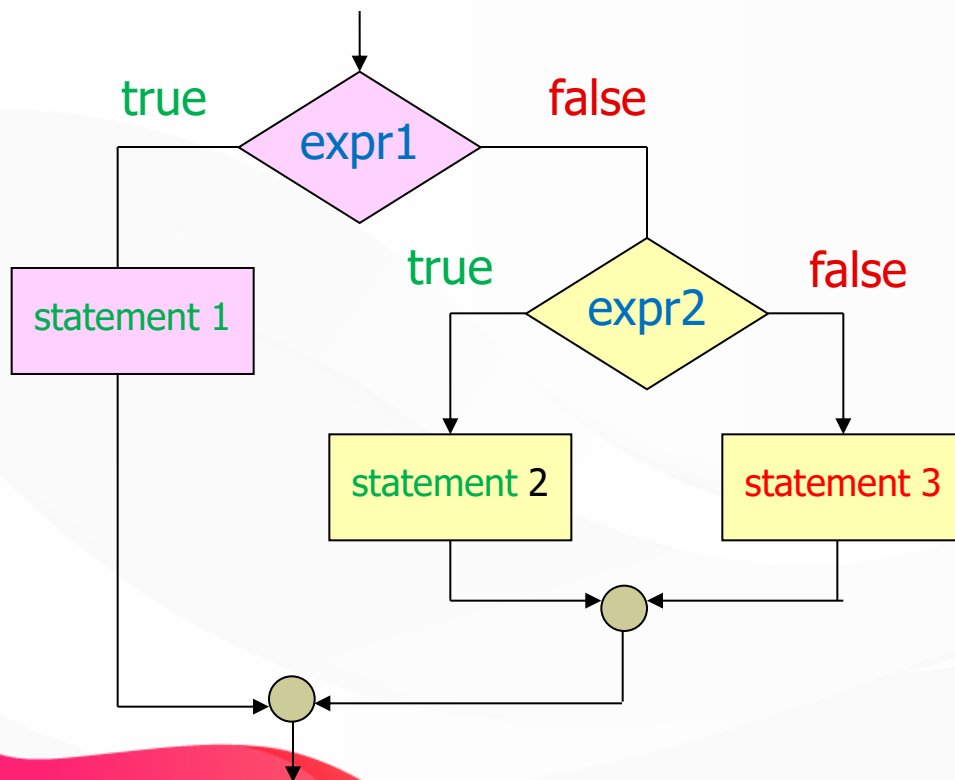
กฎพื้นฐานการใช้คำสั่ง if

การใช้ statement-if ในการเลือกทำแบบต่างๆ มีกฎที่ควรจำดังต่อไปนี้

- นิพจน์(condition) ของ if จะต้องอยู่ในวงเล็บและให้ผลลัพธ์ออกมาเป็นจริงหรือเท็จ
- ในส่วนของ if และ else ไม่ต้องมีเครื่องหมาย semicolon
- Statement หลัง if และหลัง else สามารถเป็น Statement รวมได้แต่ต้องอยู่ในเครื่องหมายปีกกา
- Statement หลัง if และ statement หลัง else สามารถสลับกันได้แต่ต้องทำ complement กับ นิพจน์(condition) ของ if

การใช้คำสั่งเลือกทำแบบ **nested if**

เป็นการใช้คำสั่งเลือกทำนี้สามารถนำหลายๆ คำสั่งมาซ้อนกันได้ ถ้าต้องการให้มีการเลือกทำหลายทางเลือก



```
if (expr 1)
    statement 1;
else
    if (expr 2)
        statement 2;
    else
        statement 3;
```

ตัวอย่างการใช้งาน Nested if

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int a,b;
5      printf("Please enter two integer: ");
6      scanf("%d %d",&a,&b);
7      if(a>b)
8          printf("%d > %d\n",a,b);
9      else
10         if(a<b)
11             printf("%d < %d\n",a,b);
12         else
13             printf("%d = %d\n",a,b);
14 }
```

ผลลัพธ์ที่ได้ ?

```
Please enter two integer: -2 9
-2 < 9
```

การควบคุมทิศทางแบบเลือกทำ if-else if

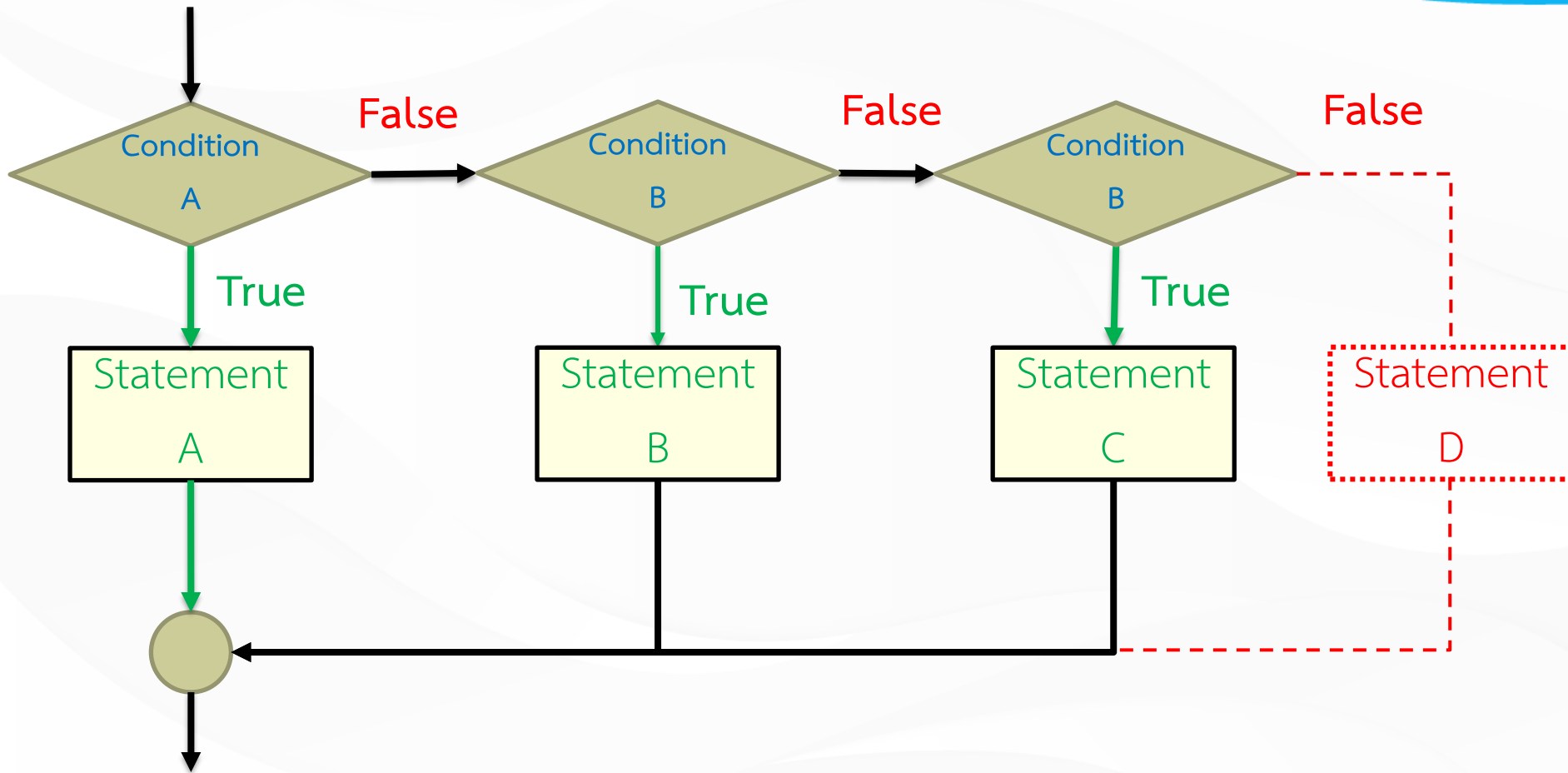
- if-else ใช้ในกรณีที่มีทางเลือกอยู่มากกว่าสองทางเลือก โดยแต่ละทางเลือกมีเงื่อนไขต่างกัน

```
if(condition_A)
    statement_A;
else if(condition_B)
    statement_B;
else if(condition_C)
    statement_C;
...
else if(condition_Y)
    statement_Y;
else
    statement_Z;
```

```
if(condition_A)
{
    statement_A1;
    ...
}
else if(condition_B)
{
    statement_B1;
    ...
}
else
{
    statement_Z;
    ...
}
```

Tip : else ตัวสุดท้ายไม่จำเป็นต้องมี

แผนภาพแสดงการทำงาน if-else if



ตัวอย่างการใช้งาน if-else if

```
1  #include <stdio.h>
2  int point;
3  int main(void)
4  {
5      printf("Enter your point : ");
6      scanf("%d",&point);
7      if((point>=80)&&(point<=100))
8          printf("Grade A\n");
9      else if((point>=70)&&(point<80))
10         printf("Grade B\n");
11     else if((point>=60)&&(point<70))
12         printf("Grade C\n");
13     else if((point>=50)&&(point<60))
14         printf("Grade D\n");
15     else
16         printf("Grade F\n");
17     return(0);
18 }
```

ตัวดำเนินการเลือกค่า

- จะมีการใช้เครื่องหมาย **question mark (?)** และเครื่องหมาย **colon (:)** กระทำกับนิพจน์สามนิพจน์ เราสามารถนำมาใช้แทนการใช้คำสั่ง if แบบง่ายๆ ได้ โดยมีรูปแบบคือ

expression 1 ? expression 2 : expression 3;

- ถ้าหากการกระทำของนิพจน์แรกเป็นจริงหรือมีค่าไม่เท่ากับศูนย์ โปรแกรมจะไปทำนิพจน์ที่สอง ถ้าเป็นเท็จจะไปทำนิพจน์ที่สาม ดังตัวอย่างต่อไปนี้

a == b ? c-- : c++;

- จากตัวอย่างถ้าหากค่าในตัวแปร **a** เท่ากับ **b** จะทำการ **ลดค่าตัวแปร c** ลง
ถ้าหากค่าในตัวแปร **a** ไม่เท่ากับ **b** จะทำการ**เพิ่มค่าตัวแปร c** ขึ้น

ตัวอย่างการใช้งาน ตัวดำเนินการเลือกค่า

การเขียนโปรแกรมตัดเกรดแบบ "ผ่าน" หรือ "ไม่ผ่าน"

```
printf("%s\n", score >= 60 ? "Passed" : "Failed");
```

ถ้า x มีค่ามากกว่า 9 จะทำให้ตัวแปร y เท่ากับ 100 แต่ถ้า x มีค่าไม่เกิน 9 จะทำให้ตัวแปร y เท่ากับ 200 ตัวดำเนินการเลือกค่าดังกล่าวจะใช้แทนคำสั่ง if ที่มีการเขียนดังนี้

```
x = 10;  
if (x > 9) y = 100;  
else y = 200;
```

ถ้าต้องการเขียนแบบตัวดำเนินการเลือกค่า ทำได้ดังนี้

```
x = 10;  
y = x > 9 ? 100 : 200;
```

ตัวอย่างการใช้งาน if-else และตัวดำเนินการเลือกค่า

จงทายตัวเลข ถ้าตัวเลขที่ใส่เข้าไปถูกต้องเครื่องจะแจ้งว่าถูกต้อง ถ้าไม่ถูกต้องเครื่องจะตอบว่าตัวเลขที่ใส่ไปมากกว่าหรือน้อยกว่าโดยใช้ตัวดำเนินการเลือกทำในโปรแกรม

```
1  int main(void)
2  {
3      int magic = 123; //กำหนดตัวเลขเป็น 123
4      int guess;
5      printf("Enter your guess : ");
6      scanf("%d", &guess);
7      if(guess == magic)
8      {
9          printf("**Right**\n");
10         printf("%d is the magic number ", magic);
11     }
12     else
13         (guess > magic)? printf("High") : printf("Low");
14     return(0);
15 }
```

ผลลัพธ์ที่ได้ ?


Enter your guess : 100 Low
Enter your guess : 321 High
Enter your guess : 123 **Right** 123 is the magic number

การควบคุมทิศทางแบบเลือกทำของ switch

switch ใช้ในกรณีมีทางเลือกให้ทำหลายทางจากเงื่อนไขร่วมกัน

```
switch(variable)
{
    case constant_A :    Statement_A1;
                        Statement_A2;
                        ...
                        break;
    case constant_B :    Statement_B1;
                        Statement_B2;
                        ...
                        break;
    case constant_C :    Statement_C1;
                        Statement_C2;
                        ...
                        break;
    ...
    default :            statement ZZ;
}
```

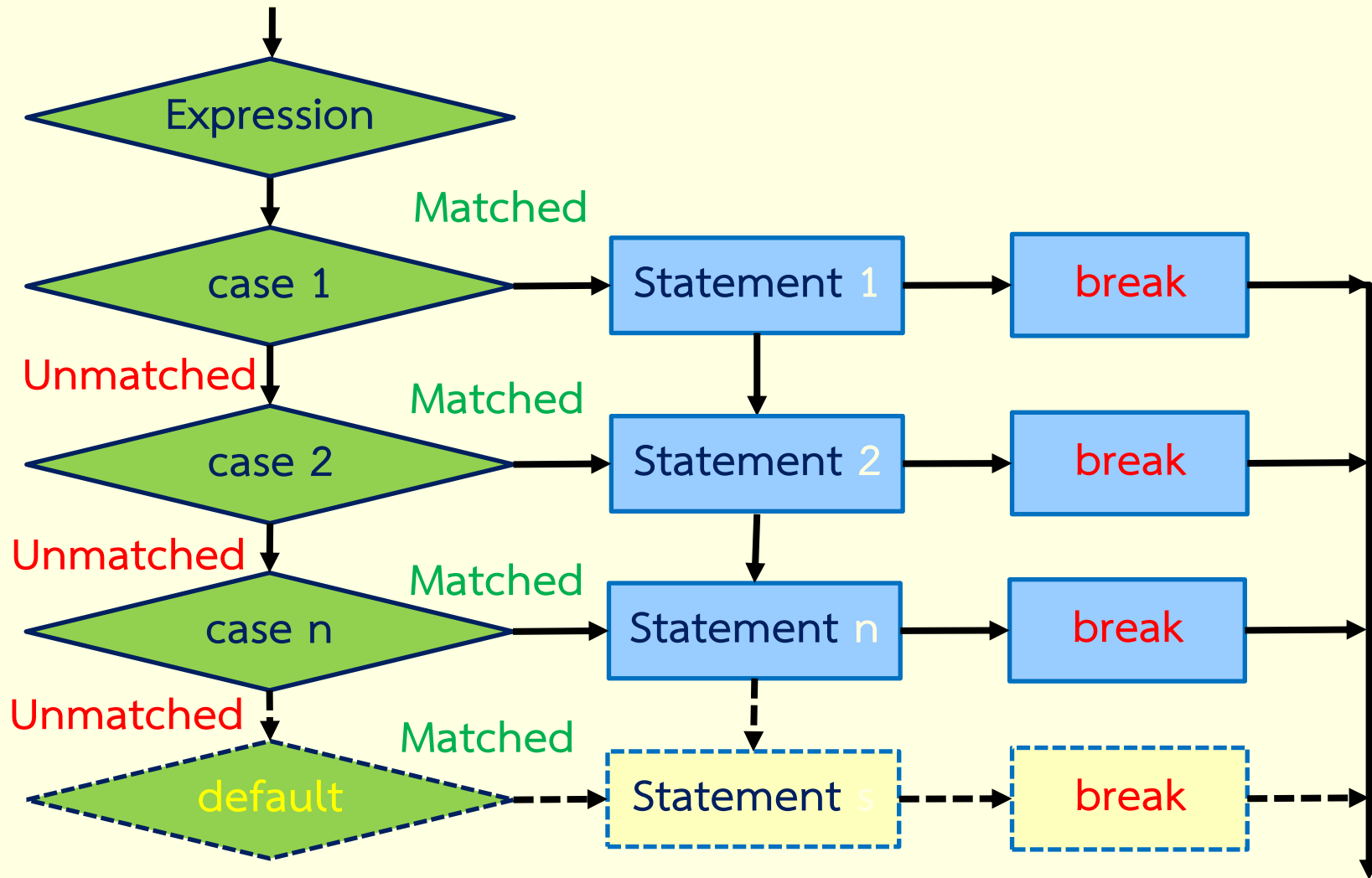
การควบคุมทิศทางแบบเลือกทำด้วยคำสั่ง switch

-  **variable** : ตัวแปรหรือนิพจน์ที่ให้ผลเป็นข้อมูลแบบ int หรือ char
- **constant_A, constant_B,...** : ค่าคงที่ชนิดเดียวกับตัวแปร variable หากค่าของ variable เท่ากับค่าคงที่ตัวใด โปรแกรมจะทำคำสั่ง case นั้น
- **break** : ใช้คั่นระหว่าง case ให้โปรแกรมออกจากทำงานของ switch หากไม่ใส่ตัวคั่น โปรแกรมจะทำการเปรียบเทียบ variable กับ constant ตัวถัดไปเรื่อยๆ
- **default** : หาก variable ไม่ตรงกับ constant ใดเลย โปรแกรมจะทำตามคำสั่งใน default

การควบคุมทิศทางแบบเลือกทำด้วยคำสั่ง switch

- คำสั่ง `if` สามารถตรวจสอบ **ความสัมพันธ์** หรือลอจิกได้
- `switch` ไม่สามารถตรวจสอบหลายๆ เงื่อนไขภายในนิพจน์เดียวกันได้
- ถ้าค่าคงที่เป็นตัวอักษร คำสั่ง `switch` จะมองเป็นเลขจำนวนเต็ม
- ค่า `default` จะมีหรือไม่ก็ได้

แผนภาพแสดงการทำงานของ switch



ตัวอย่างการใช้งาน switch

```
1  #include <stdio.h>
2  char grade;
3  int main(void)
4  {
5      printf("Enter your grade : ");
6      scanf("%c",&grade);
7      switch(grade)
8      {
9          case 'A':
10             printf("80-100\n"); break;
11          case 'B':
12             printf("70-79\n"); break;
13          case 'C':
14             printf("60-69\n"); break;
15          case 'D':
16             printf("50-59\n"); break;
17          default:
18             printf("0-49\n");
19      }
20      return(0);
21 }
```

ผลลัพธ์ที่ได้ ?

Enter your grade : A
80-100

Enter your grade : D
50-59

Enter your grade : b
0-49

ทำไมป้อน b แล้วได้ผล
0-49 ?

ตัวอย่างการใช้งาน switch

```
1  #include <stdio.h>
2  int choice;
3  float radius, circum, area;
4  int main(void)
5  {
6      printf("1.Cirumference of the circle\n");
7      printf("2.Area of the circle\n");
8      printf("Enter your choice 1 or 2 : ");
9      scanf("%d",&choice);
10     printf("Enter radius of the circle : ");
11     scanf("%f",&radius);
12     switch(choice) {
13     case 1 : circum = 2*3.14156*radius;
14             printf("Circumference of the circle =%f\n",circum);
15             break;
16     case 2 : area = 3.14156*radius*radius;
17             printf("Area of the circle = %f\n",area);
18             break;
19     }
20     return(0);
21 }
```

ผลลัพธ์ที่ได้ ?

1.Cirumference of the circle
2.Area of the circle

Enter your choice 1 or 2 : 1
คำนวณอะไร ?

Enter your choice 1 or 2 : 2
คำนวณอะไร ?

Enter your choice 1 or 2 : 3
คำนวณอะไร ?

จบบทที่ 4-1

การควบคุมทิศทางการทำงานของโปรแกรมแบบเลือกทำ
Control Structure (Selection)