

## LAST LECTURE

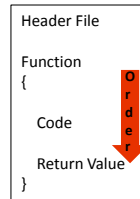
- โครงสร้างภาษาซี

```

1 // Example: Source Code
2 void main()
3 {
4     // Main function
5     printf("Basic C Program \n Displaying Text!\n");
6     // Display
7     // Program */
8     return;
9 }
10
11
12
13

```

Example: Source Code



- องค์ประกอบของภาษาซี

- ชนิดของข้อมูล และการแปลงชนิดของข้อมูล
- การประกาศตัวแปรและการให้ค่าแแรกเริ่ม

- Operator Precedence

- การแสดงผลข้อมูลผ่านทางหน้าจอ

- `printf("FormatControlString", Arguments);`
- การจัดข้อความชิดซ้ายหรือชิดขวา

- การรับข้อมูลจากผู้ใช้ผ่านคีย์บอร์ด

- `scanf("FormatControlString", Arguments);`

1

## THIS LECTURE – INTRODUCTION TO C CONTROL STRUCTURE

### Introduction to Control Structures โครงสร้างโปรแกรม

- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

2

## แนวทางออกแบบโปรแกรมเชิงโครงสร้าง

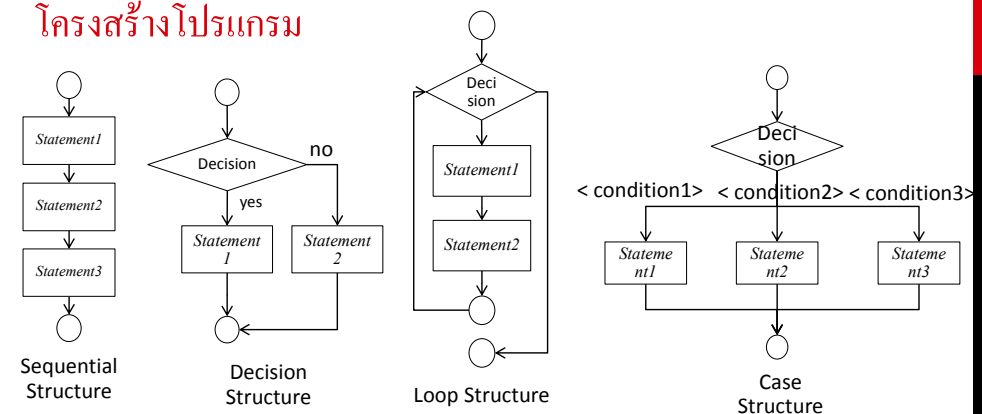
### โครงสร้างการแก้ปัญหา 4 แบบ

- Sequential Structure การทำงานแบบตามลำดับ
  - ทำคำสั่งต่อกันไปตามลำดับขั้นที่เขียนไว้
- Decision Structure การตัดสินใจทำตามเงื่อนไข
  - เปรียบเทียบเงื่อนไข เพื่อตัดสินใจทำคำสั่งที่เหมาะสม
- Loop Structure การทำซ้ำ
  - วนรอบเพื่อทำงานเดิมซ้ำ จนกว่าจะเสร็จ
- Case Structure การทำงานตามกรณี
  - นำข้อมูลมาเทียบกับกรณี แล้วทำงานตามกรณีที่เกี่ยวข้อง

3

## FLOWCHART OF CONTROL STRUCTURES

### โครงสร้างโปรแกรม

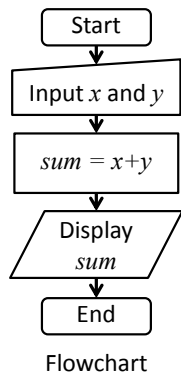


- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

4

# CONTROL STRUCTURE - SEQUENTIAL

- **Sequential** : การทำงานตามลำดับขั้นโดยไม่ข้ามหรือกระโดดไปไหน



- **Example**: โปรแกรมคำนวณผลบวกของตัวเลข 2 ตัว

```
1  #include <stdio.h>
2
3  main()
4  {
5      int x, y, sum;
6
7      printf("Input value of x: ");
8      scanf("%d",&x);
9
10     printf("Input value of y: ");
11     scanf("%d",&y);
12
13     sum=x+y;
14     printf("Result of %d + %d is: %d\n", x,y,sum);
15
16     return;
17 }
18
```

5

# CONTROL STRUCTURE

- **Introduction to Control Structures** โครงสร้างโปรแกรม

- **Sequential Structure** การทำงานแบบตามลำดับ
- **Decision Structure** การตัดสินใจทำตามเงื่อนไข
  - Statement and Logical Expression
  - If Statement
  - If-Else Statement
  - Cascaded if statement
  - Nested If
- **Loop Structure** การทำซ้ำ
- **Case Structure** การทำงานตามกรณี

6

## WHAT IS A STATEMENT?

คำสั่ง (**Statements**) คือ คำสั่งที่เขียนในโปรแกรมแต่ละบรรทัด แต่ละคำสั่งปิดด้วยเครื่องหมาย **semicolon (;)**

E.g.

```
printf("Hello!\n");
```

กลุ่มคำสั่ง (**a compound statement**) หรือ บล็อก (**block**) คือ คำสั่งหลายคำสั่งที่เขียนต่อกันเป็นชุด และทำให้เป็นกลุ่มคำสั่งด้วยการครอบด้วย **วงเล็บปีกกา (braces)**

E.g.

```
if (2 < 3)
{
    number = number + 1;
    printf("%d\n", number);
}
```

คำสั่งว่าง (**an empty statement**) คือ ไม่เขียนอะไร แต่ใส่ **semicolon** ปิดไปเฉยๆ

E.g.

```
if (condition);
```

7

## การเขียนเงื่อนไข เพื่อเทียบตรรกะ

ตัวอย่างการเขียน **Boolean expression** ในประโยค **If**

- สมมติให้ตัวแปรต่างๆ มีค่าดังนี้
  - $x = 3, y = 4, \text{max} = 100, \text{min} = 0$ , และ  $\text{ch} = 'c'$
- ผลลัพธ์ของ **Boolean expression** ต่อไปนี้คืออะไร?
  1.  $x \leq 0$
  2.  $x == y$
  3.  $\text{max} \geq \text{min}$
  4.  $\text{ch} < 'a'$
  5.  $\text{max}$
  6.  $\text{min} != 0$
  7.  $\text{max} == 99 + 1$
  8.  $\text{min} - 50 < 0$

8

## การเขียนเงื่อนไขเพื่อเทียบตรรกะ

### Logical Operators (AND &&, OR ||, NOT !)

- AND &&, OR ||, NOT ! เป็นโอเปอเรเตอร์ที่ใช้สร้างนิพจน์เพื่อเทียบตรรกะ
- ในภาษา C เราจะใช้ Logical Operators ร่วมกับ
  - โอเปอเรเตอร์เปรียบเทียบค่า เช่น <, >, <=, >=, และ == เพื่อสร้าง expression ได้
- ตัวอย่าง: สมมุติตัวแปรที่มีค่าดังนี้
  - temp = 50, MAX\_TEMP = 90, precip = 2.0, num\_votes = 20, votes\_needed = 20, และ elected = 0;
  - สองเงื่อนไขต่อไปนี้จะมีค่าผลลัพธ์เป็นอะไร?
    - (temp < MAX\_TEMP) && (precip > 0)
    - (num\_votes >= votes\_needed) || (!elected)

9

## การเขียนเงื่อนไขด้วยตรรกะตรงข้าม

### ตรรกะตรงข้าม (Complementing conditions)

เราสามารถแปลงตรรกะใดๆ เป็นตรรกะตรงข้ามได้ด้วยการใส่เครื่องหมาย **Not (!)** นำหน้าตรรกะนั้น

- ตัวอย่าง: ตรรกะตรงข้ามของ temp > 32 คือ
  - !(temp > 32)
  - temp <= 32
- ตัวอย่าง: ตรรกะตรงข้ามของ temp == 32 คือ
  - !(temp == 32)
  - (temp != 32)

10

## การเขียนเงื่อนไขเพื่อเทียบตรรกะ

ผลการเทียบเงื่อนไขเป็นไปตาม ลำดับความสำคัญของโอเปอเรเตอร์ (Operator precedence)

- ตัวแปรแต่ละตัวมีค่าดังต่อไปนี้

flag = 0, y = 4.0, z = 2.0, and x = 3.0

- จงพิจารณาเงื่อนไขต่อไปนี้

!flag || (y + z >= x - z)

Level	Operator	Associativity
1	() (parenthesis), [] (array element), . (structure member reference)	Left to right
2	- (negative sign), ++ (increment), -- (decrement), & (address-of), * (pointer indirection), sizeof (the not operator)	Right to left
3	* (multiplication), / (division), % (modulus)	Left to right
4	+ (addition), - (subtraction)	Left to right
5	< (less than), <= (less than or equal to), > (greater than), >= (greater than or equal to)	Left to right
6	= (equal to), != (not equal to)	Left to right
7	&& (logical and)	Left to right
8	(logical or)	Left to right
9	? : (the conditional operator)	Right to left
10	=, *=, /=, %*, +=, -= (assignment operators)	Right to left
11	, (the comma operator)	Left to right

11

## การเขียนเงื่อนไขเพื่อเทียบตรรกะ

### การประเมินผลเงื่อนไขทางลัด (short-circuit evaluation)

- การพิจารณาว่านิพจน์เพียงส่วนแรกก็สามารถสรุปได้ว่า นิพจน์ดังกล่าวเป็นจริงหรือเท็จ
- โอเปอเรเตอร์ && (and) ถ้าส่วนแรกของนิพจน์เป็นเท็จ ดังนั้นเงื่อนไขทั้งหมดเป็นเท็จ
  - Example: (5 < 3) && (4 > 3)
- โอเปอเรเตอร์ || (or), ถ้าส่วนแรกของนิพจน์เป็นจริง ดังนั้นเงื่อนไขทั้งหมดเป็นจริง
  - Example: (4 > 2) || (2 > 3)

CS102

12

## การเขียนเงื่อนไข เพื่อเทียบตรรกะ

เราอาจจะเอาผลการเทียบเงื่อนไขมากำหนดโดยใช้ตัวแปรชนิด `int`

เรียกว่าการกำหนดค่าด้วยผลเชิงตรรก (Logical assignment)

### • Example:

```
int temp;           /*input temperature */
int below_freezing /*temperature state*/
scanf("%d",&temp);
below_freezing = (temp < 32); /*set temp.
state*/
```

• เราสามารถใช้ตัวแปรชนิด `int` เป็นตัวเทียบเงื่อนไขได้ ตัวอย่างเช่น

```
below_freezing && (dew_point < 32)
```

13

## CONTROL STRUCTURE

### • Introduction to Control Structures โครงสร้างโปรแกรม

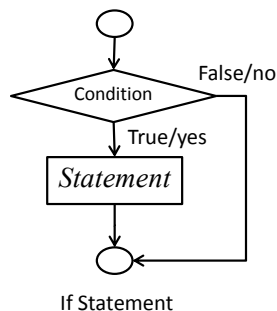
- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
  - Statement and Logical Expression
  - If Statement
  - If-Else Statement
  - Cascaded if statement and Switch
  - Nested If
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

14

## CONTROL STRUCTURE – IF STATEMENT

**Decision :** การตัดสินใจทำคำสั่งตามผลการเทียบเงื่อนไข (condition)

- หากเงื่อนไขเป็นจริง จะทำงานที่กำหนดไว้



### In Writing

```
if (Boolean Expression )
then Statement to be performed
```

### C Code

```
if ( Expression )
statement
```

**If statement** ใช้เมื่อต้องการเลือกทำงานให้ตรงตามเงื่อนไขที่ต้องการ

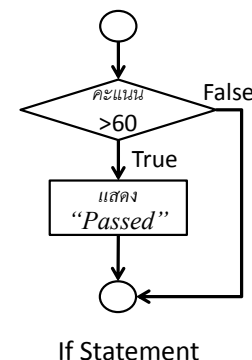
มิฉะนั้น การทำงานนั้นจะถูกข้ามไป

15

## IF STATEMENT ตัวอย่างที่ 1

- ในภาษาเขียน:

ถ้า (if) คะแนนมากกว่า 60 คะแนน (then) ให้แสดง "Passed"



ข้อมูลที่ใช้เปรียบเทียบคือ คะแนน

- ในภาษา C:  
`int point;`  
`if (point > 60)`  
`printf("Passed");`

### In Writing

```
if (Boolean Expression )
then Statement to be performed
```

### C Code

```
if ( Expression )
statement
```

16

## IF STATEMENT ตัวอย่างที่ 2

Example: oddnum.c

Test Odd Number:

ข้อมูล

number, int (input)

วิธีทำ

output "Enter an integer"

input number

if (number is odd)

then

{

output the number

}

```
#include <stdio.h>
/* Read in a number, and echo
it
if it is odd */

int main()
{

    return 0;
}
```

Cited from slides provided by Worawan Worawan Diaz Carballo

17

## IF STATEMENT ตัวอย่างที่ 2

Example: oddnum.c

Read in a number, and  
print it if it is odd

output "Enter an  
integer"

input number

if (number is odd)

then

{

output the number

}

```
#include <stdio.h>
/* Read in a number, and echo
it
if it is odd */

int main()
{
    int number;

    printf("Enter an integer:
");
    scanf("%d", &number);

    return 0;
}
```

Cited from slides provided by Worawan Worawan Diaz Carballo

18

## IF STATEMENT ตัวอย่างที่ 2

Example: oddnum.c

Read in a number, and  
print it if it is odd

output "Enter an  
integer"

input number

if (number is odd)

then

{

output the number

}

```
#include <stdio.h>
/* Read in a number, and echo
it
if it is odd. */

int main()
{
    int number;
    printf("Enter an integer:
");
    scanf("%d", &number);
    if (number % 2 != 0)
    {
        printf("%d is an odd
number\n", number);
    }
    return 0;
}
```

Cited from slides provided by Worawan Worawan Diaz Carballo

19

## IF STATEMENT ตัวอย่างที่ 2

Example: oddnum.c

Read in a number, and  
print it if it is odd

output "Enter an  
integer"

input number

output the number

อย่าได้ "then" ตรง  
นี้!

```
#include <stdio.h>
/* Read in a number, and echo
it
if it is odd. */

int main()
{
    int number;
    printf("Enter an integer:
");
    scanf("%d", &number);
    if (number % 2 != 0)
    {
        printf("%d is an odd
number\n", number);
    }
    return 0;
}
```

Cited from slides provided by Worawan Worawan Diaz Carballo

20

## NOTES ON IF

ส่วนของโปรแกรมต่อไปนี้ ให้ผลลัพธ์เหมือนกัน?

**A**

```
#include <stdio.h>

main()
{
    int Grade = 60;
    if(Grade>60)
    {
        printf("Passed\n");
    }
    printf("Nothing\n");
}
```

A Compound Statement

**B**

```
#include <stdio.h>

main()
{
    int Grade = 60;
    if(Grade>60) printf("Passed\n"); printf("Nothing\n");
}
```

A Statement

**C**

```
#include <stdio.h>

main()
{
    int Grade = 60;
    if(Grade>60)
    {
        printf("Passed\n");
        printf("Nothing\n");
    }
}
```

A Compound Statement

Cited from slides provided by Worawan Worawan Diaz Carballo

21

## NOTES ON IF

จุดผิด ที่มักจะเจอบ่อยๆ

```
if (Grade > 60);
{
    printf("Passed\n");
}
printf("Nothing\n");
```

No semi-colon here!

ถ้าใส่ ; (semi-colon) แปลว่า ไม่มีสิ่งที่ต้องทำ (empty statement)

Cited from slides provided by Worawan Worawan Diaz Carballo

22

## NOTES ON IF

จุดผิด ที่มักจะเจอบ่อยๆ (2)

Should be ==

```
if (Grade = 100)
{
    printf("Passed\n");
}
printf("Nothing\n");
```

Cited from slides provided by Worawan Worawan Diaz Carballo

23

## EXAMPLE1 – IF STATEMENT

ifRoom.c

1. ถ้าใส่inputเป็น 25 จงบอกว่า output เป็นอย่างไร
2. ถ้าใส่inputเป็น 29 จงบอกว่า output เป็นอย่างไร
3. ถ้าใส่inputเป็น 28 จงบอกว่า output เป็นอย่างไร

```
1  #include <stdio.h>
2
3  main()
4  {
5      int width;
6      printf("Enter room width: ");
7      scanf("%d", &width);
8
9      if(width<=28)
10     {
11         printf("You room is small.\n");
12         printf("Sorry this carpet is too big.\n");
13     }
14     printf("The width of your room is %d.\n", width);
15 }
16
```

24

## EXAMPLE2 – IF STATEMENT

จงเติมโปรแกรมสำหรับรับอายุจากผู้ใช้ ถ้าผู้ใช้อายุน้อยกว่า 18 ปี ให้แสดงข้อความว่า "You are too young" ให้สมบูรณ์

```
# _____ < _____ .h> 1

int main( )
{
    int age;

    printf("Enter your age: ");
    scanf("____",__); 2
    if(_____) 3
        printf("_____ \n"); 4
    printf("Your age is ____ \n", ____); 5
    return 0;
}
```

ถ้าใส่ age = 18 ลงไป output ของโปรแกรมจะเป็นอย่างไร **6**

25

## CONTROL STRUCTURE

### • Introduction to Control Structures โครงสร้างโปรแกรม

- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
  - Statement and Logical Expression
  - If Statement
  - If-Else Statement
  - Cascaded if statement and Switch
  - Nested If
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

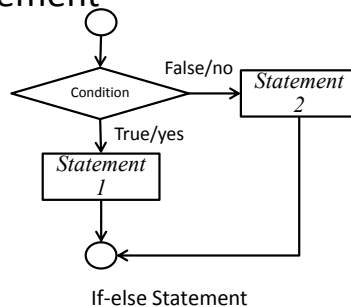
26

## IF-ELSE STATEMENT

- Else Statement ต้องใช้กับ If Statement
- หากเงื่อนไขเป็นจริง จะทำงาน Statement 1
- เมื่อเงื่อนไขเป็นเท็จ จะทำงาน Statement 2

### In Writing

**if** (Boolean Expression)  
**then** คำสั่งที่ทำเมื่อผลเป็นจริง  
**else** คำสั่งที่ทำเมื่อผลเป็นเท็จ

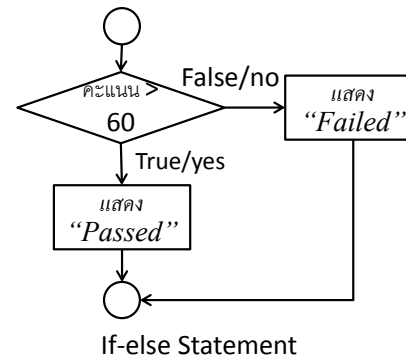


### C Code

```
if ( expression )
    statement1
else
    statement2
```

27

## IF-ELSE STATEMENT ตัวอย่างที่ 1



### C Code

```
if ( expression )
    statement1
else
    statement2
```

- ในภาษาเขียน:

ถ้า (if) คะแนนมากกว่า 60 คะแนน ให้แสดง "Passed"

(else) มิฉะนั้นให้แสดง "Failed"

ข้อมูลที่ใช้เปรียบเทียบคือ คะแนน

- ในภาษา C:

```
int point;
if ( point > 60 )
    printf("Passed");
else
    printf("Failed");
```

28

## IF-ELSE STATEMENT ตัวอย่างที่ 2

Example: oddeven.c

Read in a number, and determine if it's odd or even

output "Enter an integer" input number

```
if (number is odd)
then
{
    output: number " is an odd number"
}
else
{
    output: number " is an even number"
}
```

```
#include <stdio.h>

/* Determine whether an input
number
is odd or even. */
main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    if (number % 2 != 0)
    {
        printf("%d is an odd
number\n",
        number);
    }
    else
    {
        printf("%d is an even
number\n",
        number);
    }
}
```

Cited from slides provided by Worawan Worawan Diaz Carballo

29

## EXAMPLE3 – IF-ELSE STATEMENT

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b;
6      int min, max;
7
8      printf("Enter value of a: ");
9      scanf("%d", &a);
10     printf("Enter value of b: ");
11     scanf("%d", &b);
12
13     if (a > b)
14     {
15         printf("Not in order\n");
16         min = b;
17         max = a;
18     }
19     else
20     {
21         printf("In order\n");
22         min = a;
23         max = b;
24     }
25     printf("Minimum is %d and Maximum is %d\n", min, max);
26     return 0;
27 }
```

จงบอกoutputของโปรแกรม

1. ถ้า a = 2 และ b = 3
2. ถ้า a = 8 และ b = 4
3. ถ้า a = 2 และ b = 2

30

## CONTROL STRUCTURE

### • Introduction to Control Structures โครงสร้างโปรแกรม

- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
  - Statement and Logical Expression
  - If Statement
  - If-Else Statement
  - Cascaded if statement and Switch
  - Nested If
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

31

## CASCADED IF STATEMENT

การเขียน If-else ต่อกัน ไปเรื่อยๆ

แต่ละblockมีการกำหนดเงื่อนไขของมันเอง

ถ้าเงื่อนไขของblockนั้นเป็นจริง statements ของเงื่อนไขนั้น จะถูกกระทำ มีเพียงเงื่อนไขเดียวเท่านั้นที่ถูกต้อง

In Writing

```
ถ้า (คะแนนมากกว่า 80)
then ได้เกรด A
มิฉะนั้น ถ้า(คะแนนมากกว่า 70 )
then ได้เกรด B
มิฉะนั้น ถ้า(คะแนนมากกว่า 60 )
then ได้เกรด C
มิฉะนั้น คุณต้องรีเกรด
```

In Writing

```
if (Boolean Expression1)
then คำสั่งที่ทำเมื่อผลเป็นจริง
else if (Boolean Expression2)
then คำสั่งที่ทำเมื่อผลเป็นจริง
else if (Boolean Expression3)
then คำสั่งที่ทำเมื่อผลเป็นจริง
else คำสั่งที่ทำเมื่อไม่ตรงเงื่อนไขใด
```

32



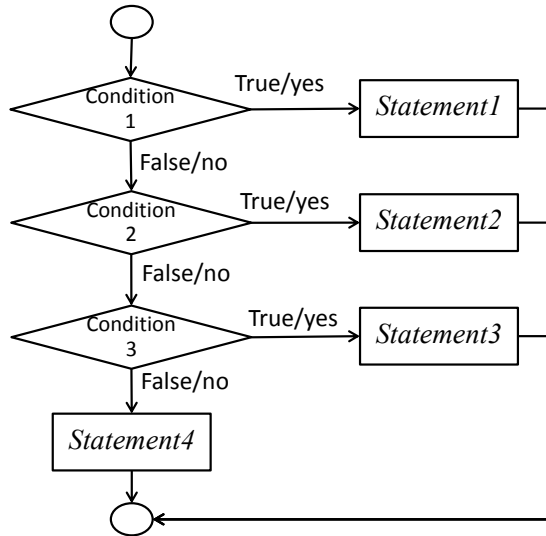
## CASCADED IF STATEMENT

### In Writing

```
if (Boolean Expression1)
then คำสั่งที่ทำเมื่อผลเป็นจริง
else if (Boolean Expression2)
then คำสั่งที่ทำเมื่อผลเป็นจริง
else if (Boolean Expression3)
then คำสั่งที่ทำเมื่อผลเป็นจริง
else คำสั่งที่ทำเมื่อไม่ตรงเงื่อนไขใด
```

### C Code

```
if ( expression1 )
    statement1
else if ( expression2 )
    statement2
else if ( expression3 )
    statement3
else
    statement4
```



Cascaded if Statement

33

## CASCADED IF STATEMENT ตัวอย่างที่ 1

- ในภาษาเขียน:

ถ้า(if)คะแนนมากกว่า80คะแนน

แสดง "Grade = A"

มิฉะนั้นถ้า(else if)คะแนนมากกว่า70

คะแนน แสดง "Grade = B"

มิฉะนั้นถ้า(else if)คะแนนมากกว่า60

คะแนน แสดง "Grade = C"

(else)มิฉะนั้นให้

แสดง "You must re-grade"

ข้อมูลที่ใช้เปรียบเทียบคือ คะแนน

- ในภาษา C:

int point; ข้อมูล

if (point > 80) เงื่อนไข

printf("Grade = A\n");

else if (point > 70) เงื่อนไข

printf("Grade = B\n");

else if (point > 60) เงื่อนไข

printf("Grade = C\n");

else

printf("You must re-grade\n");

34

## EXAMPLE4 – CASCADED IF STATEMENT

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int points;
6
7      printf("Enter your points: ");
8      scanf("%d", &points);
9
10     if (points >= 80)
11     {
12         printf("Your grade is A\n");
13     }
14     else if (points >= 70)
15     {
16         printf("Your grade is B\n");
17     }
18     else if (points >= 60)
19     {
20         printf("Your grade is C\n");
21     }
22     else
23     {
24         printf("You must re-grade. See you next year.\n");
25     }
26     return 0;
27 }
```

จงบอกoutputของโปรแกรม

1. ถ้า points = 90

2. ถ้า points = 59

3. ถ้า points = 60

36

## CONTROL STRUCTURE

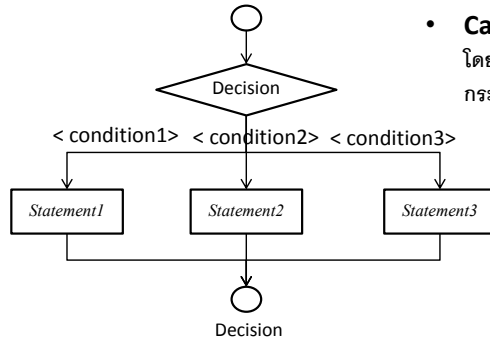
- Introduction to Control Structures โครงสร้างโปรแกรม

- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
  - Statement and Logical Expression
  - If Statement
  - If-Else Statement
  - Cascaded if statement and Switch (Case Structure)
  - Nested If
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

37

## CASE STRUCTURE การทำงานตามกรณี

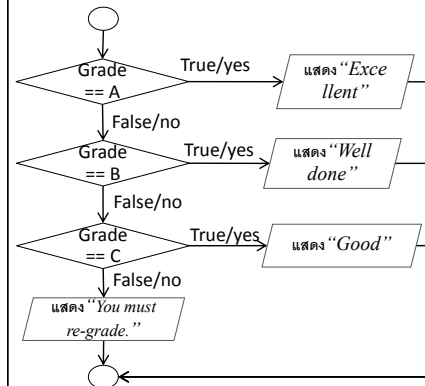
- ปัญหาอย่างหนึ่งของการเขียน **if** แบบ **cascaded if** คืออ่านเข้าใจยาก
- ยิ่งซ้อนหลายชั้นเท่าไร ยิ่งอ่านเข้าใจยากขึ้นเท่านั้น
- โอกาสที่จับ **If-else** ผิดคู่ ("mis-nest" if statements) มีสูง
- ดังนั้น หากตัวเงื่อนไขของการเขียน **cascaded if** มาจากการเทียบค่าตัวแปรเพียงตัวเดียว
  - ใช้คำสั่ง **switch-cases** จะดีกว่า
  - Switch-case** เป็น Case Structure ซึ่งมีการทำงานตามกรณี



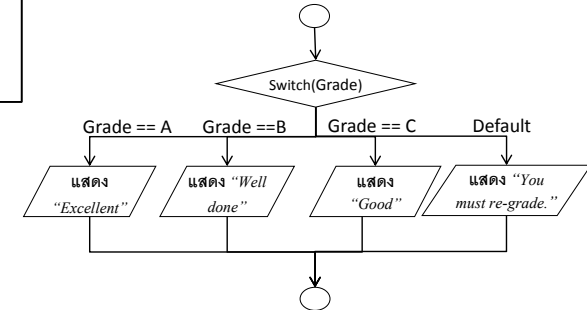
- Case Structure:** แต่ละเงื่อนไขจะถูกเปรียบเทียบโดยใช้ตัวแปรหนึ่งตัวแปร หากเงื่อนไขเป็นจริง การกระทำที่อยู่ในเงื่อนไขนั้น ๆ จะถูกกระทำ

38

## CASE STRUCTURE การทำงานตามกรณี – SWITCH STATEMENT



Cascaded if Statement



Case Structure

## ตัวอย่างการใช้ SWITCH

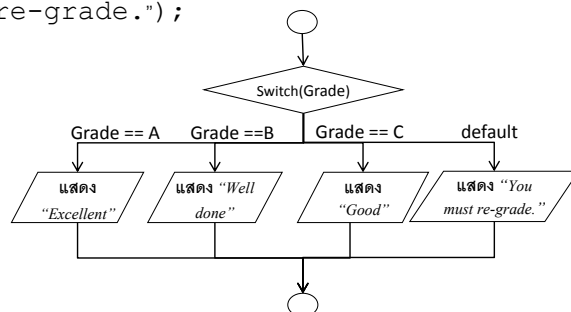
**switch (Grade)**

```
{
    case 'A': printf("excellent!"); break;
    case 'B': printf("well done!"); break;
    case 'C': printf("good!"); break;
    default:
        printf("You must re-grade.");
};
```

Key

กระโดดออกเมื่อทำ  
กรณี 'A' เสร็จแล้ว

หากไม่ตรงกับเงื่อนไข  
ใด จะไปที่กรณีนี้



Case Structure

40

## SWITCH STATEMENTS (ต่อ)

ข้อจำกัดของ **switch**

- Switch** ไม่สามารถใช้ได้กับตัวแปรที่มีชนิดของข้อมูลที่มีชนิดเป็น **double** หรือ **string**:
- ตัวอย่างเช่น
  - `char code[20] = "home run";`  
`switch (code) { ... /* can't do this */`
  - `double era = 1.67;`  
`switch (era) { ... /* can't do this either */`

ควรกำหนด **default** ทุกครั้ง

A common mistake is to forget to end a case with **break**. If a **break** is forgotten, execution "falls" through to the next case label! (C can be cruel.)

```
switch (Grade)
{
    case 'A': printf("excellent!");
    case 'B': printf("well done!");
    case 'C': printf("good!"); break;
    default:
        printf("You must re-grade.");
};
```

No break;

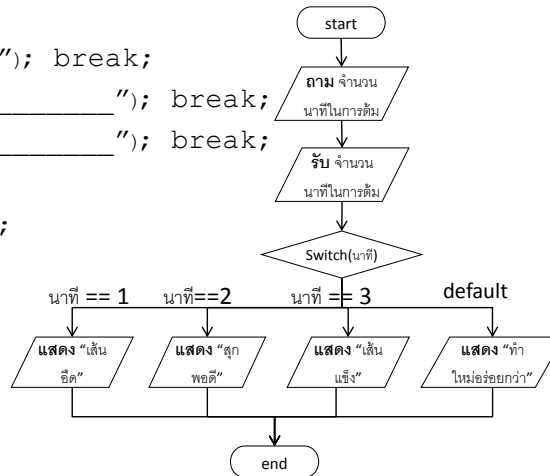
CS102

41

## จงเดิมให้โปรแกรมสมบูรณ์ตาม FLOWCHART

switch (\_\_\_\_)

```
{
  case __: printf("เส้นอืด"); break;
  case __: printf("_____"); break;
  case __: printf("_____"); break;
  default:
    printf("ทำใหม่หรือร่อยกว่า");
};
```



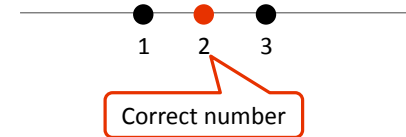
42

## EXAMPLE5 – CASCADED IF STATEMENT

Q:

Guessing a number:

GuessNum.c



```
if (num >= 3)
{
  printf("Too big\n");
}
else if (num == 2)
{
  printf("Correct!\n");
}
else if (num <= 1)
{
  printf("Too small\n");
}
```

43

## EXAMPLE6 – CASCADED IF STATEMENT

Letter2.c

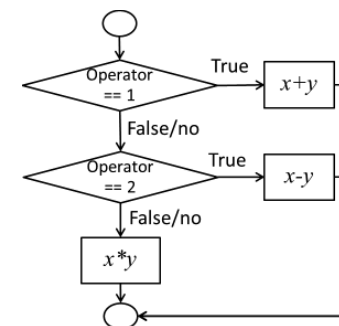
```
if (ch >= 'a' && ch <= 'z')
{
  printf("%c is in lower case.\n", ch);
}
else if (ch >= 'A' && ch <= 'Z')
{
  printf("%c is in upper case.\n", ch);
}
else if (ch >= '0' && ch <= '9')
{
  printf("%c is a digit with value %d.\n", ch, ch - '0');
}
```

44

## EXAMPLE CASCADED IF AND SWITCH

จงเขียนโปรแกรมที่รับเครื่องหมายการคำนวณและตัวเลขที่ใช้ในการคำนวณเข้ามา และแสดงผลการคำนวณ

กำหนดให้ 1 แทนค่า เครื่องหมาย +  
กำหนดให้ 2 แทนค่า เครื่องหมาย -  
กำหนดให้ 3 แทนค่า เครื่องหมาย \*



Output "Select an Operator:

1 for +, 2 for -, and 3 for \*

Input operator

Output "Enter the first number"

Input x (int)

Output "Enter the second number"

Input y (int)

if (operator is 1)

then

```
{
  output: "Result x + y is" x+y
}
```

else if (operator is 2)

```
{
  output: "Result x - y is" x-y
}
```

else

```
{
  output: "Result x * y is" x*y
}
```

45

## CASCADED IF

```
int num1, num2, oper, result;

printf("Select 1: for \"+\"\n");
printf("Select 2: for \"-\"\n");
printf("Select 3: for \"*\"\n");
printf("Select an operator:\n");
scanf("%d", &oper);

printf("Enter the first number:");
scanf("%d", &num1);

printf("Enter the second number:");
scanf("%d", &num2);
```

Calculator\_CascadedIf.c

```
if(oper == 1)
{
    result = num1+num2;
    printf("%d + %d = %d", num1, num2, result);
}
else if(oper == 2)
{
    result = num1-num2;
    printf("%d - %d = %d", num1, num2, result);
}
else if (oper == 3)
{
    result = num1*num2;
    printf("%d * %d = %d", num1, num2, result);
}
else
{
    printf("Wrong operator\n");
}
```

เปลี่ยนโค้ดที่ให้มาให้เป็นการ  
เขียนโดยใช้ Switch Case

46

## CONCLUSION

- **Introduction to Control Structures** โครงสร้างโปรแกรม
  - Sequential Structure การทำงานแบบตามลำดับ
  - Decision Structure การตัดสินใจทำตามเงื่อนไข
    - Statement and Logical Expression
    - If Statement
    - If-Else Statement
    - Cascaded if statement and Switch (Case Structure)
    - Nested If
  - Loop Structure การทำซ้ำ
  - Case Structure การทำงานตามกรณี

47