

w06-Lec

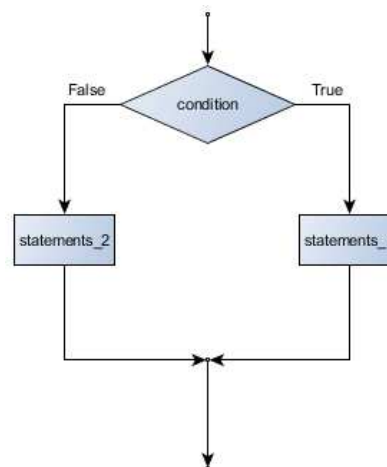
Conditional Control Statement

Part III

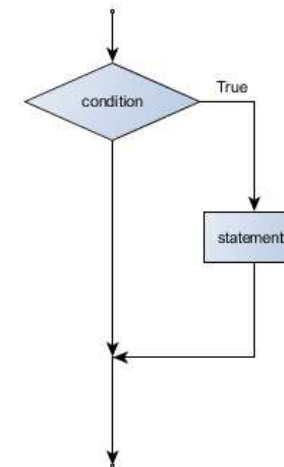
Assembled for 204111
by Kittipitch Kuptavanich

Conditional Executions

- **Basic form:**
(if - else)

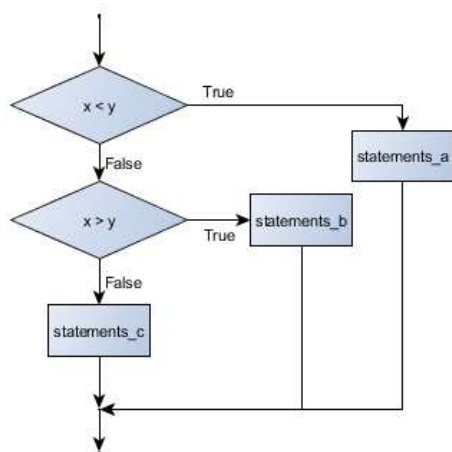


- **Omitting else**

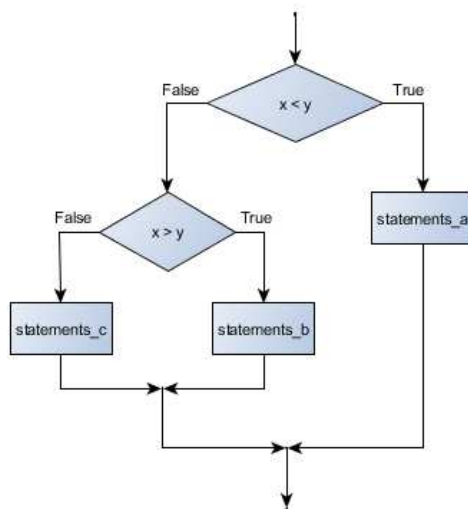


Conditional Executions [2]

- **Chain Conditionals:**
(if - else if)



- **Nested Conditionals:**



Nested Conditionals

- ในบางกรณี เราสามารถใช้ basic conditionals, nested conditionals หรือ chained conditionals เพื่อแก้ปัญหาเดียวกันได้ (เช่น max-mid-min)
- อย่างไรก็ตามการเขียนโปรแกรม nested condition (if ซ้อน if) นั้น มีลักษณะโครงสร้างที่อ่านและเข้าใจได้ยากกว่า conditional statement ลักษณะอื่น
- ควรหลีกเลี่ยงหากเป็นไปได้

Nested Conditionals [2]

- เราสามารถนำ Logical Operators มาประยุกต์ใช้เพื่อหลีกเลี่ยงการใช้ nested condition ได้เช่น

```
if (0 < x) {           // Assume x is an int here
    if (x < 10) {
        printf("x is a positive single digit.");
    }
}
```

- จะเห็นได้ว่า function `printf()` จะถูกเรียกใช้ก็ต่อเมื่อเงื่อนไขใน `if` เป็นจริงทั้ง 2 กรณี

- เราสามารถใช้อะไร `&&` operator เพื่อสร้าง expression ใหม่ ดังนี้

```
if ((0 < x) && (x < 10)) { // no nested if
    printf("x is a positive single digit.");
}
```

5

Logical Opposites [2]

- พิจารณา code

```
if (!(age >= 18)) {
    printf("You're too young to get a driving license!");
}
```

- สามารถ simplify ได้เป็น

```
if (age < 18) { //อ่านและเข้าใจได้ง่ายกว่า
    printf("You're too young to get a driving license!");
}
```

7

Logical Opposites

- ใน relational operators แต่ละตัว จะมีเครื่องหมายตรงกันข้ามอยู่ ตัวอย่างเช่น

- ในประเทศไทย ผู้ที่ต้องการทำใบขับขี่ จะต้องมียุมากกว่าหรือเท่ากับ 18 ปี (≥ 18)

- ไม่สามารถทำใบขับขี่ได้หากมีอายุต่ำกว่า 18 ปี (< 18)
- สังเกตว่า เครื่องหมายตรงกันข้ามของ \geq คือ $<$

operator	logical opposite
<code>==</code>	<code>!=</code>
<code>!=</code>	<code>==</code>
<code><</code>	<code>>=</code>
<code><=</code>	<code>></code>
<code>></code>	<code><=</code>
<code>>=</code>	<code><</code>

6

De Morgan's Laws

- เช่นเดียวกับใน Boolean Algebra เราสามารถนำ De Morgan's Laws มาใช้เพื่อ simplify ประโยคเงื่อนไข (conditional statements) ได้ดังนี้

$$\sim(a \wedge b) = \sim a \vee \sim b$$

$$\sim(a \vee b) = \sim a \wedge \sim b$$



$$!(x \ \&\& \ y) == (!x \ || \ !y)$$

$$!(x \ || \ y) == (!x \ \&\& \ !y)$$

8

De Morgan's Laws [2]

Example:

- วัยเรียน (หรือนักศึกษา) คือคนที่มีอายุในช่วง 6 ปีขึ้นไป จนถึง 21 ปี
 - $6 \leq \text{age} \leq 21 \rightarrow \text{student}$
 - $((\text{age} \geq 6) \ \&\& \ (\text{age} \leq 21)) \rightarrow \text{student}$
- คนที่มีอายุ น้อยกว่า 6 ปี หรือ มากกว่า 75 ปี ไม่ถือเป็นวัยทำงาน
 - $\text{age} < 6$ หรือ $\text{age} > 75 \rightarrow \text{not professional}$
 - $((\text{age} < 6) \ || \ (\text{age} > 75)) \rightarrow \text{not professional}$
 - $!((\text{age} < 6) \ || \ (\text{age} > 75)) \rightarrow \text{professional}$

9

Refactoring Conditionals

```
if ((age >= 6) && (age <= 21)) {
    printf("student\n");
}
else if ((age >= 6) && (age <= 75)) {
    printf("professional\n");
}
```

- สามารถใช้กฎการกระจายเพื่อดึง $(\text{age} \geq 6)$ ออกมา

```
if (age >= 6) {
    if (age <= 21){
        printf("student\n");
    }
    else if (age <= 75) {
        printf("professional\n");
    }
}
```

- Readability?

Note: ตัวอย่างนี้เป็นไปเพื่อการแสดงการใช้สมบัติของ Boolean Algebra

11

De Morgan's Laws [3]

Example (contd.):

```
if ((age >= 6) && (age <= 21)) {
    printf("student\n");
}
else if (!(age < 6) || (age > 75)) {
    printf("professional\n");
}
```

- เมื่อใช้ De Morgan's Law จะได้

```
if ((age >= 6) && (age <= 21)) {
    printf("student\n");
}
else if ((age >= 6) && (age <= 75)) {
    printf("professional\n");
}
```

10

Refactoring Conditionals [2]

Love6 Game Revisited:

- กำหนด integer 2 ตัว **first** และ **second** โปรแกรมจะแสดงผล **true** ก็ต่อเมื่อ
 - ตัวใดตัวหนึ่งมีค่า 6
 - ผลบวกของทั้งสองตัวมีค่า 6
 - ผลต่างของทั้งสองตัวมีค่า 6

```
if ((first == 6) || (second == 6)) ||
    (first + second == 6) ||
    ((first - second == 6) || (second - first == 6))) {
    printf("true");
} else {
    printf("false");
}
```

12

Refactoring Conditionals [3]

Love6 Game Revisited (2):

ในกรณีที่เงื่อนไขมีลักษณะเป็น mutually exclusive (เหตุการณ์ไม่เกิดร่วม) เราสามารถเปลี่ยน OR statement ให้เป็น else if ได้

```
if ((first == 6) || (second == 6)) {
    printf("true");
} else if (first + second == 6) {
    printf("true");
} else if ((first - second == 6) || (second - first == 6)) {
    printf("true");
} else {
    printf("false");
}
```

13

Tips

- ในกรณีโครงสร้างแบบ chain (else if) ที่ input สามารถ evaluate เป็น True ได้มากกว่าหนึ่งเงื่อนไข เงื่อนไขแรกที่เป็น True เท่านั้นที่จะถูกดำเนินการ

```
int x = 360;

if (x % 2 == 0) {
    printf("2 divides x");
} else if (x % 3 == 0) {
    printf("3 divides x");
} else if (x % 5 == 0) {
    printf("5 divides x");
} else {
    printf("x is not divisible by 2 3 or 5");
}
```

15

Refactoring Conditionals [4]

Love6 Game Revisited (3):

- เปลี่ยน OR statement ที่เหลือให้เป็น else if

```
if (first == 6) { //case 1a
    printf("true");
} else if (second == 6) { //case 1b
    printf("true");
} else if (first + second == 6) { //case 2
    printf("true");
} else if (first - second == 6) { //case 3a
    printf("true");
} else if (second - first == 6) { //case 3b
    printf("true");
} else {
    printf("false");
}
```

14

Example 1: Leap Year

- ปีอธิกสุรทินโดยปกติแล้ว จะเป็นปีคริสต์ศักราช ที่หารด้วย 4 ลงตัว เช่นปี 2012, 2016 และ 2020 Case 1
- ข้อยกเว้น (ในกรณีปีหาร 4 ลงตัว)
 - ปีที่หารด้วย 100 ลงตัว ไม่ใช่ปีอธิกสุรทิน (1700, 1800, 1900) Case 2
 - แต่ปีที่หารด้วย 400 ลงตัว เป็นปีอธิกสุรทิน (1600, 2000) Case 3
- STEP1:** สร้าง test case

	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

16

Example 1: Leap Year

- **STEP2:** พิจารณา condition แรก (case 1 และ 4) ก่อน

- ปีคริสตศักราช ที่หารด้วย 4 ลงตัว เช่นปี 2012, 2016 และ 2020

```
if (year % 4 == 0) {
    printf("yes");
} else {
    printf("no");
}
```

	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

17

Example 1: Leap Year [2]

- **STEP3:** พิจารณา ใส่ข้อยกเว้นแรก (Case 2)

- ข้อยกเว้น (ในกรณีปีหาร 4 ลงตัว)

- ปีที่หารด้วย 100 ลงตัว ไม่ใช่ปีอธิกสุรทิน (1700, 1800, 1900)

```
if (year % 4 == 0) {
```

```
    printf("yes")
```

```
    } else {
        printf("no");
    }
```

	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

18

Example 1: Leap Year [3]

- **STEP3:** พิจารณา ใส่ข้อยกเว้นแรก (Case 2)

- ข้อยกเว้น (ในกรณีปีหาร 4 ลงตัว)

- ปีที่หารด้วย 100 ลงตัว ไม่ใช่ปีอธิกสุรทิน (1700, 1800, 1900)

```
if (year % 4 == 0) { //ต้องเพิ่มเป็น nested เพราะอยู่เป็นกรณีย่อยของ case 1
    if (year % 100 == 0) {
        printf("no");
    } else {
        printf("yes")
    }
} else {
    printf("no");
}
```

	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

19

Example 1: Leap Year [4]

- **STEP4:** พิจารณา ใส่ข้อยกเว้นที่สอง (Case 3)

- ข้อยกเว้น (ในกรณีปีหาร 4 ลงตัว)

- ปีที่หารด้วย 100 ลงตัว ไม่ใช่ปีอธิกสุรทิน (1700, 1800, 1900)

- แต่ปีที่หารด้วย 400 ลงตัว เป็นปีอธิกสุรทิน (1600, 2000)

```
if (year % 4 == 0) {
    if (year % 100 == 0) {
        printf("no");
```

```
    } else {
        printf("yes")
    }
```

```
    } else {
        printf("no");
    }
```

	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

20

Example 1: Leap Year [5]

STEP4: พิจารณา ใส่ข้อยกเว้นที่สอง (Case 3)

- ข้อยกเว้น (ในกรณีปีหาร 4 ลงตัว)
 - ☒ ปีที่หารด้วย 100 ลงตัว ไม่ใช่ปีอธิกสุรทิน (1700, 1800, 1900)
 - ☑ แต่ปีที่หารด้วย 400 ลงตัว เป็นปีอธิกสุรทิน (1600, 2000)

```
if (year % 4 == 0) {
    if (year % 100 == 0) {
        printf("no");
    } else if (year % 400 == 0) {
        printf("yes")
    } else {
        printf("yes")
    }
} else {
    printf("no");
}
```

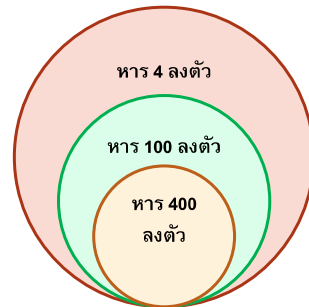
	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

21

Example 1: Leap Year [7]

STEP6: Reviewing

```
if (year % 4 == 0) {
    if (year % 400 == 0) {
        printf("yes");
    } else if (year % 100 == 0) {
        printf("no")
    } else {
        printf("yes")
    }
} else {
    printf("no");
}
```



	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

กรณี condition เป็น subset ซึ่งกันและกัน (ไม่แยกจากกันเด็ดขาด) ให้สร้าง condition จากกรณีที่เจาะจงกว่าก่อน (เล็กไปหาใหญ่)

Practice: หากต้องการเปลี่ยนรูปแบบเป็น chain ล้วนๆ แทน nested if จะทำอย่างไร

23

Example 1: Leap Year [6]

STEP5: Testing

```
if (year % 4 == 0) {
    if (year % 100 == 0) {
        printf("no");
    } else if (year % 400 == 0) {
        printf("yes")
    } else {
        printf("yes")
    }
} else {
    printf("no");
}
```

วิธีแก้: สลับตำแหน่งเงื่อนไข

- Year = 2400 ได้ output = ?
- Why?
- จะแก้ bug อย่างไร?

	Test Case	Result
Case 1	2012 2016 2020	YES
Case 2	1700 1800 1900	NO
Case 3	1600 2000 2400	YES
Case 4	2013 2014 2015	NO

22

References

- <http://openbookproject.net/thinkcs/python/english3e/conditionals.html>

24