

# CHAPTER04

การเขียนโปรแกรมแบบมีเงื่อนไข (Condition Programming)

# บทนำ

- ในการเขียนโปรแกรมในลักษณะที่ง่ายที่สุดก็คือการเขียนโปรแกรมแบบเรียงลำดับ ดังที่ได้ศึกษาใน **chapter03** ที่ผ่านมา
- แต่การเขียนโปรแกรมที่มีความซับซ้อนมากขึ้น โปรแกรมแบบเรียงลำดับจะไม่สามารถตอบโจทย์ปัญหาเหล่านี้ได้
- ดังนั้นจึงต้องมีการศึกษาการเขียนโปรแกรมแบบมีเงื่อนไขด้วย

# ข้อมูลสตริง และการจัดการรูปแบบข้อมูลชนิดตัวเลข

- การประกาศใช้ตัวแปรแบบสตริง มีรูปแบบ ดังนี้

```
String strName = "Tom Cruise";
```

- จากรูปแบบที่แสดงตัวแปร **strName** จะถูกกำหนดค่าให้มีค่าเท่ากับ **Tom Cruise** โดย **String** เป็นชนิดข้อมูลที่เรียกว่า **class** ในภาษาจาวา ซึ่งคุณสมบัติต่างๆ ของคลาสสตริงจะได้ศึกษาต่อไป

# เมธอด `length()`

- ❑ เมธอด `length()` เป็นเมธอดที่ใช้สำหรับหาความยาวของข้อความ หรือสตริง สามารถเรียกใช้งานได้ดังนี้

```
variable_name.length();
```

- ❑ โดยที่ **variable name** คือชื่อตัวแปรสตริง
- ❑ ตัวอย่างการใช้งาน

```
String stdName = "Tom Hanks";  
System.out.println("ชื่อมีความยาว = " + stdName.length());
```

# เมธอด substring()

- ❑ เมธอด **substring()** ใช้ทำหน้าที่แยกข้อความ มีรูปแบบการใช้ดังนี้

```
variable_name.substring(begin,end);
```

- ❑ โดยที่ **begin** คือตำแหน่งเริ่มต้นในการแยกข้อความ สำหรับตำแหน่งข้อความจะเริ่มนับจากตำแหน่งที่ 0
- ❑ **end** คือตำแหน่งสุดท้ายที่ต้องการแยกข้อความออกมา ซึ่งตำแหน่งสุดท้ายของข้อความ คือตำแหน่งที่อยู่หลังข้อความนั้น

# เมธอด `substring()`

- ❑ ตัวอย่างการใช้งานเมธอด `substring()`

```
String stdName = "Tom Hanks";  
String sub = stdName.substring(4,9);
```

- ❑ จากตัวอย่างนี้ผลลัพธ์ที่ได้จากการแยกข้อความ (`substring`) คือ `"Hanks"` ก็เลยต้องให้ `begin` มีค่าเป็นตำแหน่งที่ 4 อยู่ตำแหน่งอักษร `'H'` ส่วน `end` นั้นจะอยู่หลังตัวอักษร `'s'` ซึ่งก็จะได้ตำแหน่งที่ 9 นั่นเอง

# การเปรียบเทียบข้อมูลสตริง

- ❑ ในกรณีที่ต้องการนำสตริงมาเปรียบเทียบกับว่าเท่ากันหรือไม่นั้น เราจะไม่สามารถใช้ตัวดำเนินการการเปรียบเทียบเท่ากัน (==) ได้
- ❑ สตริงมีเมธอดที่ใช้สำหรับเปรียบเทียบการเท่ากันของสตริงได้แก่
- ❑ เมธอด **equals()** ใช้เปรียบเทียบการเท่ากันของสตริง โดยผลลัพธ์ของเมธอด **equals()** จะมีค่าเป็น **true** และ **false** เท่านั้น
- ❑ เมธอด **compareTo()** ใช้เปรียบเทียบค่าของสตริงว่าเท่ากันหรือไม่ โดยผลลัพธ์มีได้ 3 ค่า คือ มากกว่า 0 , เท่ากับ 0 และน้อยกว่า 0

# เมธอด equals()

- ❑ เขียนโปรแกรมต่อไปนี้แล้วรันหาผลลัพธ์ที่ได้

```
1 |
2 public class TestMethodEquals {
3     public static void main(String[] args) {
4         String str1 = "Tom Hanks";
5         String str2 = "Tom & Jerry";
6         System.out.println(str1.equals(str2));
7     }
8 }
```

- ❑ ผลลัพธ์

@ Javadoc Problems Declaration Console

<terminated> TestMethodEquals [Java Application] C:\Program Files  
false

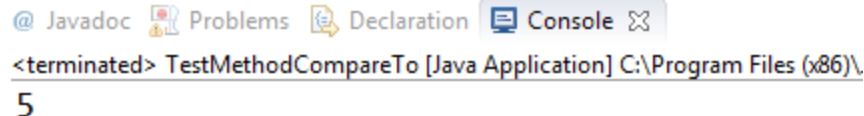


# เมธอด compareTo()

- เขียนโปรแกรมต่อไปนี้แล้วค้นหาผลลัพธ์ที่ได้

```
1 |  
2 public class TestMethodCompareTo {  
3     public static void main(String[] args) {  
4         String str1 = "Tom Hanks";  
5         String str2 = "Tom Cruise";  
6         System.out.println(str1.compareTo(str2));  
7     }  
8 }  
9
```

- ผลลัพธ์



The screenshot shows an IDE interface with tabs for Javadoc, Problems, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The text in the console is:   
<terminated> TestMethodCompareTo [Java Application] C:\Program Files (x86)\...  
5

# คำสั่งที่ใช้สำหรับการเขียนโปรแกรมแบบมีเงื่อนไข

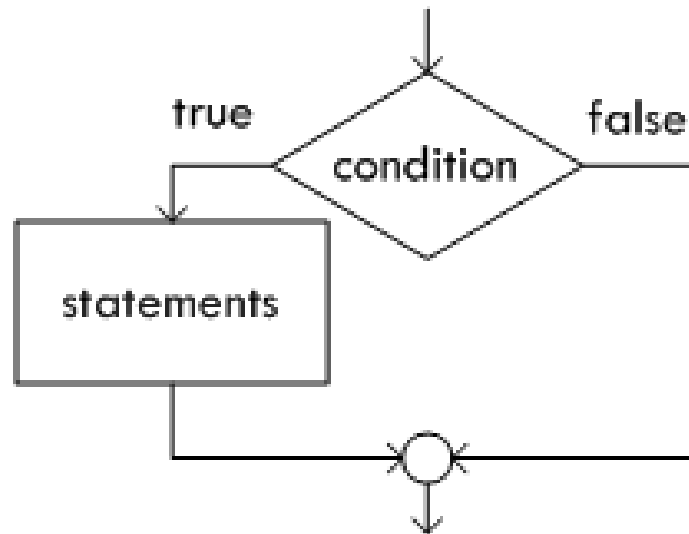
- ในภาษาจาวามีคำสั่งเงื่อนไขอยู่ด้วยกัน 4 คำสั่ง ดังนี้
- 1. คำสั่ง `if()`
- 2. คำสั่ง `if() else`
- 3. คำสั่ง ~~nested~~ multi-way `if() else`
- 4. คำสั่ง `switch() case`
- 5. `condition operator ( ? : )`

# 1. คำสั่ง if()

□ รูปแบบคำสั่ง if()

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

□ ผังงานคำสั่ง if()



# คำสั่ง `if()`

- ❑ การทำงานของคำสั่ง `if()` เริ่มจากตรวจสอบเงื่อนไข (**condition**)
- ❑ โดยเงื่อนไขนั้นอาจจะได้จากการเปรียบเทียบ เช่น นิพจน์ทางด้านการเปรียบเทียบ  $a > 30$ ,  $a == 30$  เป็นต้น
- ❑ ผลลัพธ์จากการเปรียบเทียบนี้มีค่าได้ 2 ค่าคือ **true** และ **false**
- ❑ โดยถ้า **condition** เป็นจริง (**true**) จะทำ **statement** (คำสั่งใดๆ)
- ❑ แต่ถ้า **condition** เป็นเท็จ (**false**) จะไม่ทำคำสั่งใดๆ ข้ามไปทำคำสั่งถัดไปทีี่ต่อจากคำสั่ง `if()`

# ตัวอย่างการใช้งานคำสั่ง `if()`

- ❑ เขียนโปรแกรมเพื่อหาผลสอบของนักเรียนแต่ละคน โดยรับชื่อ และคะแนนสอบของนักเรียนทางแป้นพิมพ์ โดยมีเงื่อนไขคือ หากคะแนนสอบมีค่าน้อยกว่า **60** ถือว่าไม่ผ่านเกณฑ์ จากนั้นให้ส่งพิมพ์ทั้งชื่อ คะแนนสอบ และผลสอบทางจอภาพ

## กำหนดตัวแปร

`stdName`      แทน ชื่อนักเรียน

`score`            แทน คะแนนสอบ

`result`            แทน ผลสอบ

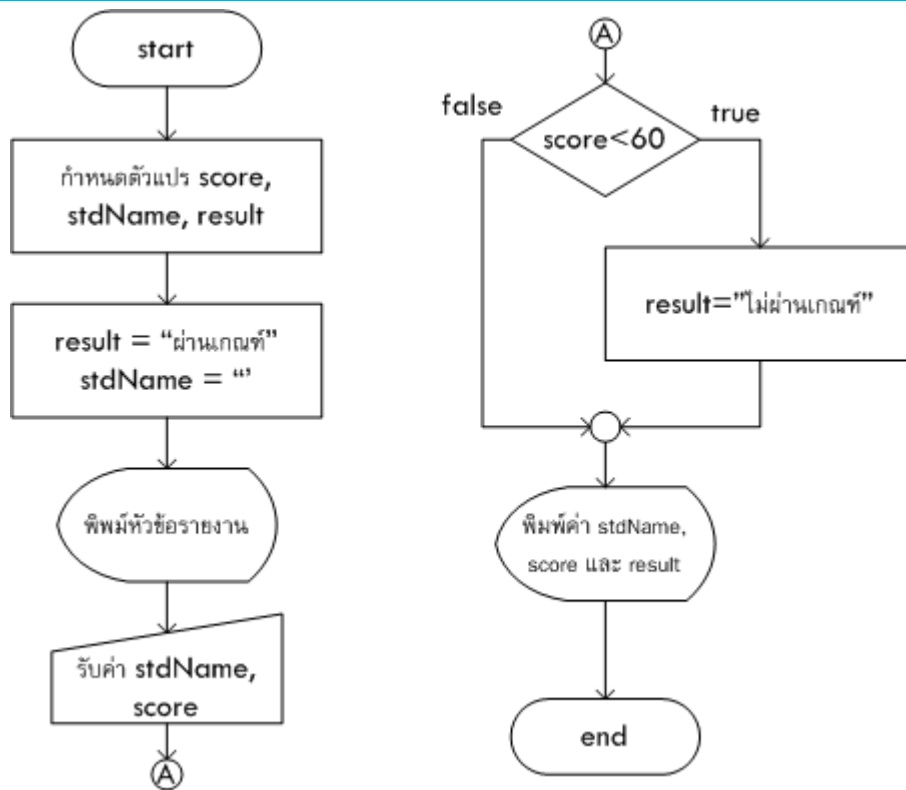
# ตัวอย่างการใช้งานคำสั่ง `if()`

## อัลกอริทึม

1. กำหนดค่าเริ่มต้นให้ผลสอบ (**result**) เท่ากับ “ผ่านเกณฑ์”
2. รับค่าชื่อนักเรียน (**stdName**) และคะแนนสอบ (**score**)
3. ทดสอบคะแนนสอบ  $\text{score} < 60$  เป็นจริงหรือเท็จ
  - 3.1 ถ้าจริง ให้ผลสอบเท่ากับ “ไม่ผ่านเกณฑ์” แล้วไปทำข้อ 4
  - 3.2 ถ้าเท็จ ให้ไปทำข้อ 4
4. พิมพ์ชื่อนักเรียน คะแนนสอบ และผลสอบ
5. จบการทำงาน

# ตัวอย่างการใช้งานคำสั่ง if()

ผังงาน



# ตัวอย่างการใช้งานคำสั่ง if()

□ โค้ด

```
1 import java.util.Scanner;
2 public class ScorePass {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int score;
6         String stdName = "";
7         String result = "ผ่านเกณฑ์";
8         System.out.println("===โปรแกรมตัดเกรด===");
9         System.out.print("กรุณาป้อนชื่อนักเรียน : ");
10        stdName = sc.nextLine();
11        System.out.print("กรุณาป้อนคะแนนสอบ : ");
12        score = sc.nextInt();
13        if(score<60)
14            result = "ไม่ผ่านเกณฑ์";
15        System.out.println("ชื่อ : " + stdName);
16        System.out.println("คะแนนสอบ : " + score);
17        System.out.println("ผลสอบที่ได้ : " + result);
18    }
19 }
```



# ตัวอย่างการใช้งานคำสั่ง if()

## ❑ ผลลัพธ์

```
@ Javadoc Problems Declaration Console
<terminated> ScorePass [Java Application] C:\Program Files (x86)\Java\jre7
===โปรแกรมตัดเกรด===
กรุณาป้อนชื่อนักเรียน : ปาน ธนพร
กรุณาป้อนคะแนนสอบ : 78
ชื่อ : ปาน ธนพร
คะแนนสอบ : 78
ผลสอบที่ได้ : ผ่านเกณฑ์
```

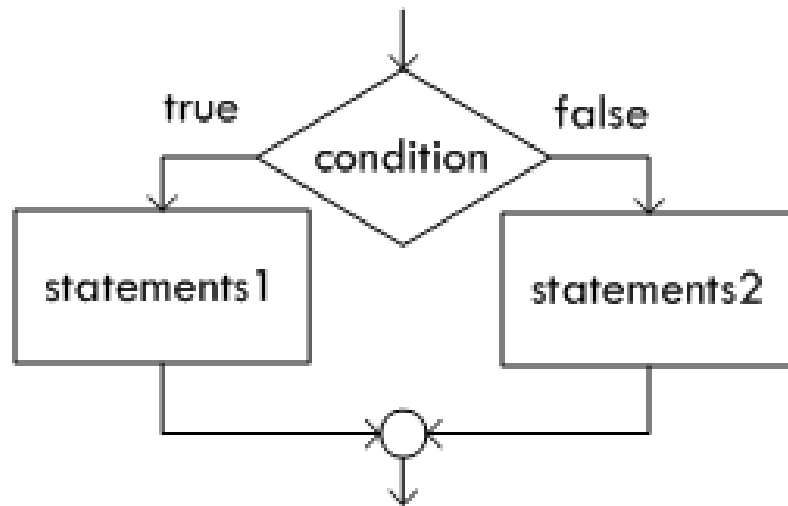
```
@ Javadoc Problems Declaration Console
<terminated> ScorePass [Java Application] C:\Program Files (x86)\Java\jre7
===โปรแกรมตัดเกรด===
กรุณาป้อนชื่อนักเรียน : แจ็ค ธนพล
กรุณาป้อนคะแนนสอบ : 57
ชื่อ : แจ็ค ธนพล
คะแนนสอบ : 57
ผลสอบที่ได้ : ไม่ผ่านเกณฑ์
```

## 2. คำสั่ง if() else

□ รูปแบบคำสั่ง if() else

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

□ ฟังก์ชัน if() else



# คำสั่ง if() else

- การทำงานของคำสั่ง **if() else** เริ่มจากตรวจสอบเงื่อนไข (**condition**)
- ถ้าเงื่อนไขเป็นจริง (**true**) จะทำ **statements 1**
- แต่ถ้าเงื่อนไขเป็นเท็จ (**false**) จะทำ **statements 2**
- สังเกตเห็นได้ว่าเงื่อนไขจะเป็นจริงหรือเท็จก็จะมีคำสั่งให้ทำ บางทีก็เรียกคำสั่ง **if() else** ว่าเป็นคำสั่งที่มี 1 เงื่อนไข 2 ทางเลือก

# ตัวอย่างการใช้งานคำสั่ง `if() else`

- เขียนโปรแกรมป้อนเลขจำนวนเต็มหนึ่งจำนวนจากแป้นพิมพ์ แล้วแสดงผลลัพธ์ว่าตัวเลขจำนวนเต็มนั้นเป็นเลขคู่ หรือเลขคี่

## กำหนดตัวแปร

**num**                    แทน เลขจำนวนเต็มที่ป้อนจากแป้นพิมพ์

**result**                แทน ผลลัพธ์จากการประมวลผลตัวเลข

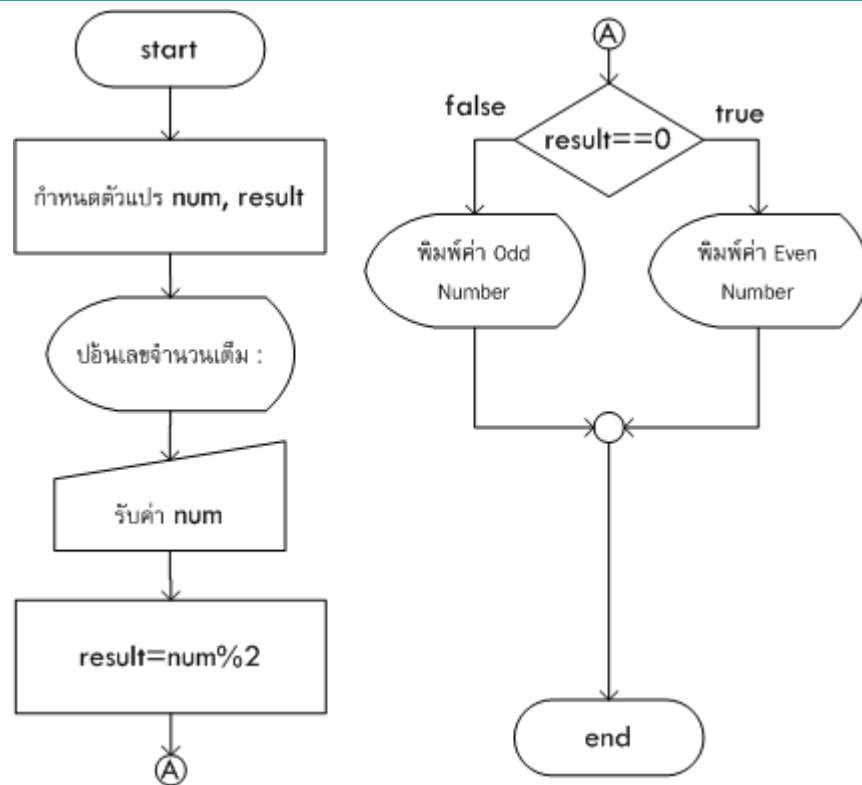
# ตัวอย่างการใช้งานคำสั่ง `if() else`

## อัลกอริทึม

1. รับเลขจำนวนเต็มจากแป้นพิมพ์เก็บไว้ในตัวแปร `num`
2. นำ `num` ไปหารด้วย 2 แล้วเก็บเศษที่ได้ในตัวแปร `result` ดังนี้  
$$\text{result} = \text{num} \% 2$$
3. ตรวจสอบเงื่อนไข
  - 3.1 ถ้า `result == 0` เป็นจริง แสดงข้อความ Even Number
  - 3.2 ถ้า `result == 0` เป็นเท็จ แสดงข้อความ Odd Number
4. จบการทำงาน

# ตัวอย่างการใช้งานคำสั่ง `if() else`

□ ผังงาน



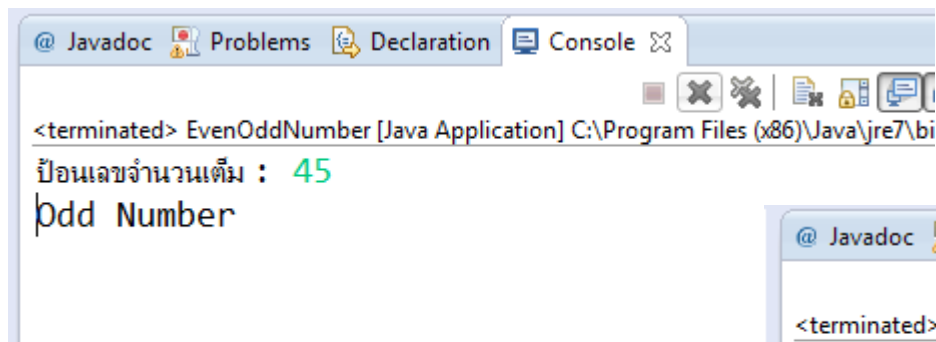
# ตัวอย่างการใช้งานคำสั่ง if() else

□ โค้ด

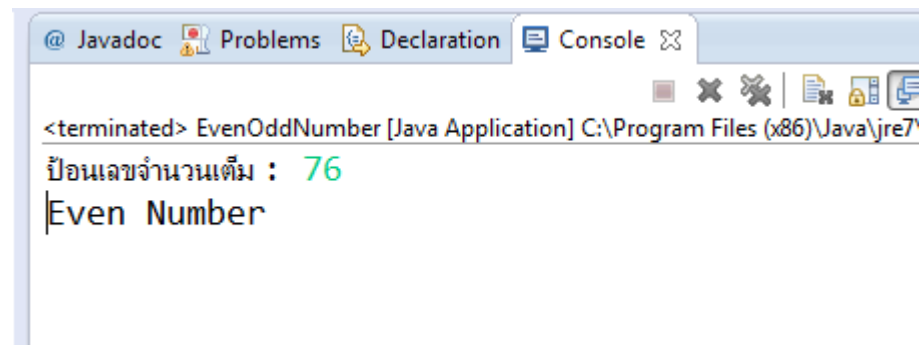
```
EvenOddNumber.java
1 import java.util.Scanner;
2 public class EvenOddNumber {
3     public static void main(String[] args) {
4         int num, result;
5         Scanner sc = new Scanner(System.in);
6         System.out.print("ป้อนเลขจำนวนเต็ม : ");
7         num = sc.nextInt();
8         result = num%2;
9         if(result==1){
10             System.out.println("Odd Number");
11         } else {
12             System.out.println("Even Number");
13         }
14         sc.close();
15     }
16 }
17
```

# ตัวอย่างการใช้งานคำสั่ง if() else

## □ ผลลัพธ์



```
@ Javadoc Problems Declaration Console  
<terminated> EvenOddNumber [Java Application] C:\Program Files (x86)\Java\jre7\bin\java.exe  
ป้อนเลขจำนวนเต็ม : 45  
Odd Number
```



```
@ Javadoc Problems Declaration Console  
<terminated> EvenOddNumber [Java Application] C:\Program Files (x86)\Java\jre7\bin\java.exe  
ป้อนเลขจำนวนเต็ม : 76  
Even Number
```



# Leap Year

- ปีไหนที่เดือนกุมภาพันธ์มี 29 วัน ปีนั้นจะเรียกว่า “Leap Year”
- วิธีการหา Leap Year มีเงื่อนไข ดังนี้
- เอาปี ค.ศ. หารด้วย 4 ลงตัว ปีนั้นจะเป็น Leap Year
- แต่มีข้อยกเว้นว่า เลขปี ค.ศ. ที่หารด้วย 4 ลงตัวนี้ ต้องหารด้วย 100 ไม่ลงตัว ยกเว้นกรณีที่หารด้วย 400 ลงตัว

# Leap Year (Cont.)

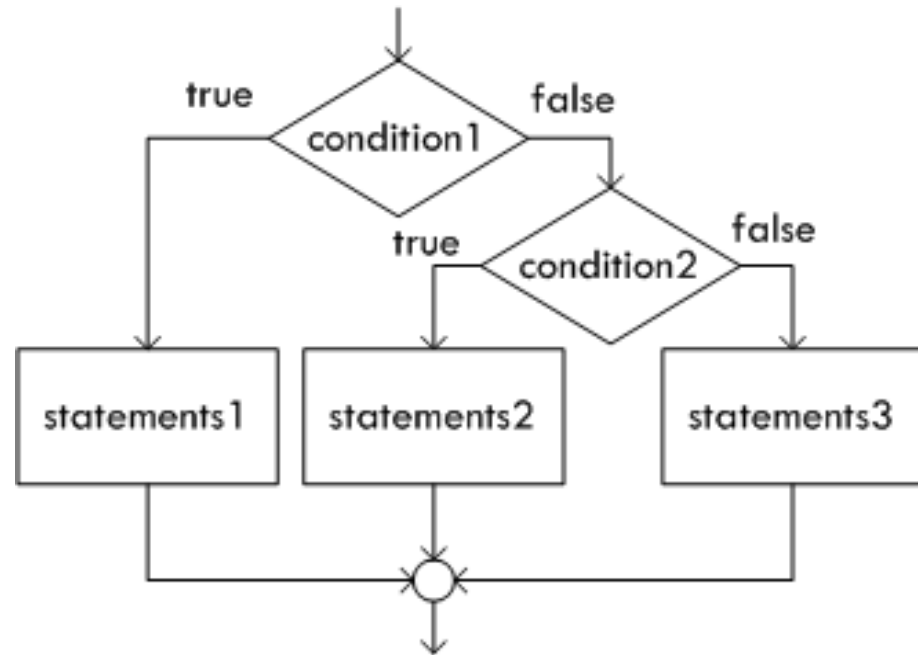
- ❑ เดือนกุมภาพันธ์ ปี ค.ศ. 2000 มี 29 วัน (เพราะหารด้วย 400 ลงตัว)
- ❑ เดือนกุมภาพันธ์ ปี ค.ศ. 1900 มี 28 วัน (เพราะหารด้วย 100 ลงตัว แต่หารด้วย 400 ไม่ลงตัว)
- ❑ เดือนกุมภาพันธ์ ปี ค.ศ. 2008 มี 29 วัน (เพราะหารด้วย 4 ลงตัว และหารด้วย 100 ไม่ลงตัว) เป็นต้น

# คำสั่ง ~~nested~~ multi-way if() else

□ รูปแบบคำสั่ง

```
if(condition1) {  
    statements1;  
} else if(condition2){  
    statements2;  
} else {  
    statements3;  
}
```

□ ผังงาน



# คำสั่ง ~~nested~~ multi-way if() else

- การทำงานของคำสั่ง ~~nested~~ multi-way if() else
- เริ่มจากตรวจสอบเงื่อนไขที่ 1 (condition1) ถ้า เป็นจริงจะทำ statements1 แล้วออกจากคำสั่ง ~~nested~~ multi-way if() else
- แต่ถ้าเงื่อนไขที่ 1 เป็นเท็จ จะทำการตรวจสอบเงื่อนไขที่ 2 (condition2) ถ้า เป็นจริงจะทำคำสั่ง statements2 แล้วออกจากคำสั่ง ~~nested~~ multi-way if() else
- แต่ถ้าเงื่อนไขที่ 2 เป็นเท็จก็จะไปทำ statements3 แล้วออกจากคำสั่ง ~~nested~~ multi-way if() else

# คำสั่ง ~~nested~~ multi-way if() else

- คำสั่ง ~~nested~~ multi-way if() else นี้สามารถมีเงื่อนไขได้มากกว่า 2 เงื่อนไข แต่ที่แสดงในตัวอย่างกำหนดไว้ที่ 2 เงื่อนไข
- ข้อสังเกต ให้จำไว้ว่าการทำงานของคำสั่ง ~~nested~~ multi-way if() else จะเหมือนกับคำสั่ง if() else ทุกประการ เพียงแต่คำสั่ง ~~nested~~ multi-way if() else จะมีเงื่อนไขได้มากกว่า 1 เงื่อนไขนั่นเอง

# ตัวอย่างการใช้คำสั่ง ~~nested~~ multi-way if() else

- เขียนโปรแกรมคำนวณหาเกรดนักเรียน จากคะแนนที่รับเข้ามาจากแป้นพิมพ์ โดยมีเงื่อนไขในการตัดเกรด ดังนี้

คะแนนตั้งแต่ 0-49 ได้เกรด F

คะแนนตั้งแต่ 50-59 ได้เกรด D

คะแนนตั้งแต่ 60-69 ได้เกรด C

คะแนนตั้งแต่ 70-79 ได้เกรด B

คะแนนตั้งแต่ 80-100 ได้เกรด A

# ตัวอย่างการใช้คำสั่ง ~~nested~~ multi-way if() else

กำหนดตัวแปร

score            แทน คะแนนสอบ

grade            แทน เกรด

อัลกอริทึม

1. กำหนดตัวแปร **score, grade**
2. รับค่า **score** จากแป้นพิมพ์

# ตัวอย่างการใช้คำสั่ง `nested if() else`

## อัลกอริทึม

1. กำหนดตัวแปร `score`, `grade`

2. รับค่า `score` จากแป้นพิมพ์

3. ตรวจสอบค่า `score`

- 3.1 ถ้า `score <= 100` และ `score >= 80` จริง กำหนดให้ `grade = "A"` แล้วไปทำข้อ 4



# ตัวอย่างการใช้คำสั่ง ~~nested~~ multi-way if() else

3.2 ถ้า  $\text{score} \leq 79$  และ  $\text{score} \geq 70$  จริง กำหนดให้  $\text{grade} = \text{"B"}$  แล้วไปทำข้อ 4

3.3 ถ้า  $\text{score} \leq 69$  และ  $\text{score} \geq 60$  จริง กำหนดให้  $\text{grade} = \text{"C"}$  แล้วไปทำข้อ 4

3.4 ถ้า  $\text{score} \leq 59$  และ  $\text{score} \geq 50$  จริง กำหนดให้  $\text{grade} = \text{"D"}$  แล้วไปทำข้อ 4

3.5 ถ้า  $\text{score} \leq 49$  และ  $\text{score} \geq 0$  จริง กำหนดให้  $\text{grade} = \text{"F"}$  แล้วไปทำข้อ 4

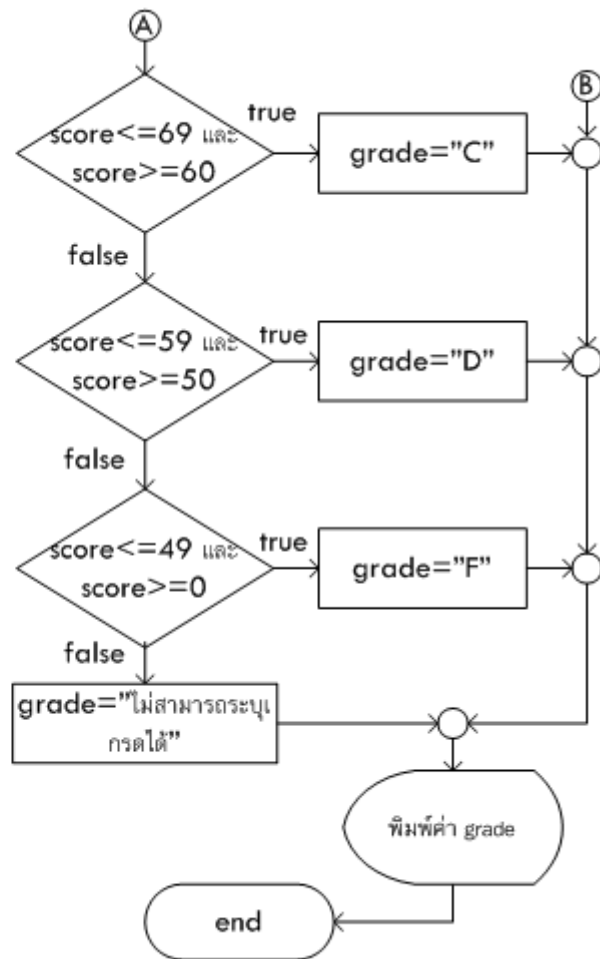
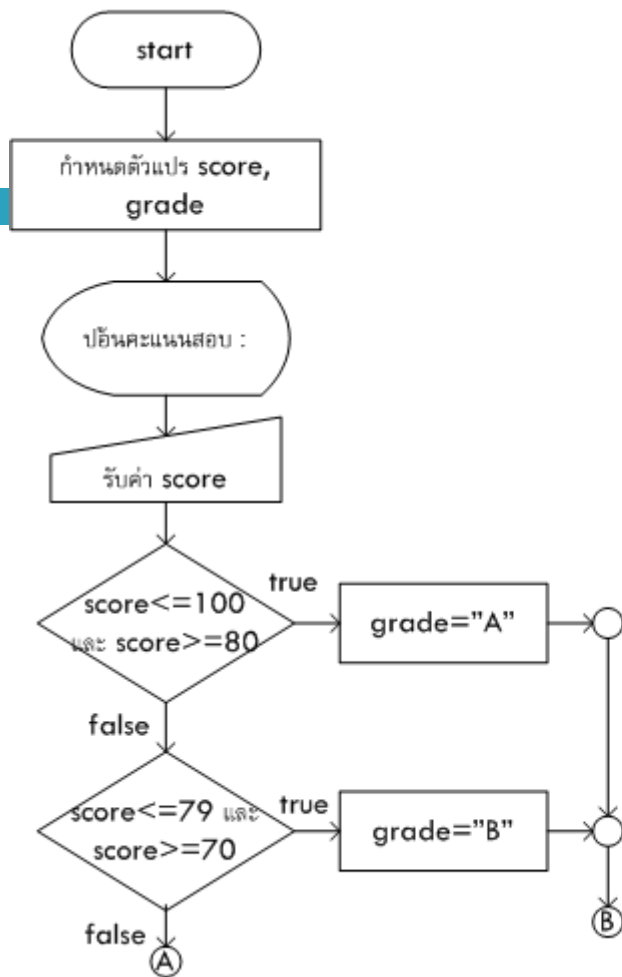
3.6 นอกนั้นกำหนดให้  $\text{grade} = \text{"ไม่สามารถระบุเกรดได้"}$

4. สั่งพิมพ์ค่า  $\text{grade}$

5. จบการทำงาน

# ตัวอย่าง

□ ผังงาน



# ตัวอย่าง

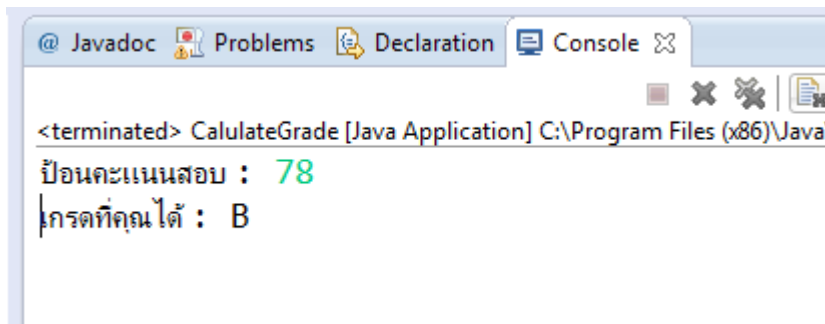
□ โค้ด

CalculateGrade.java

```
1 import java.util.Scanner;
2 public class CalculateGrade {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int score;
6         String grade;
7         System.out.print("ป้อนคะแนนสอบ : ");
8         score = sc.nextInt();
9         if(score<=100 && score>=80)
10             grade = "A";
11         else if(score<=79 && score>=70)
12             grade = "B";
13         else if(score<=69 && score>=60)
14             grade = "C";
15         else if(score<=59 && score>=50)
16             grade = "D";
17         else if(score<=49 && score>=0)
18             grade = "F";
19         else
20             grade = "ไม่สามารถระบุเกรดได้";
21         System.out.println("เกรดที่คุณได้ : " + grade);
22         sc.close();
23     }
24 }
```

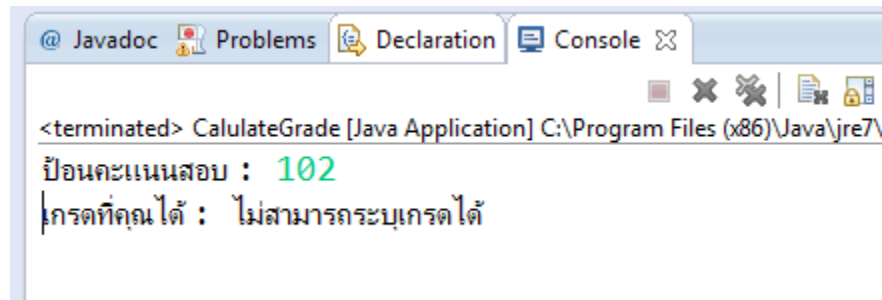
# ตัวอย่างการใช้คำสั่ง ~~nested~~ multi-way if() else

## □ ผลลัพธ์



The screenshot shows a Java IDE console window with the following tabs: @ Javadoc, Problems, Declaration, and Console. The console output is as follows:

```
<terminated> CalculateGrade [Java Application] C:\Program Files (x86)\Java\jre7\bin\java.exe  
ป้อนคะแนนสอบ : 78  
เกรดที่คุณได้ : B
```



The screenshot shows a Java IDE console window with the following tabs: @ Javadoc, Problems, Declaration, and Console. The console output is as follows:

```
<terminated> CalculateGrade [Java Application] C:\Program Files (x86)\Java\jre7\bin\java.exe  
ป้อนคะแนนสอบ : 102  
เกรดที่คุณได้ : ไม่สามารถระบุเกรดได้
```

## 4. คำสั่ง switch() case

□ รูปแบบคำสั่ง

```
switch(variable) {  
    case value1 : statements1; break;  
    case value2 : statements2; break;  
    case value3 : statements3; break;  
        :  
    case valueN : statementsN; break;  
    default : statements;  
}
```

# คำสั่ง switch() case

- ❑ หลักการทำงานของคำสั่ง switch() case
- ❑ นำค่าในตัวแปร (**variable**) ไปตรวจสอบกับ value1-valueN ว่ามีค่าเท่ากับค่าใด
- ❑ สมมติว่าตัวแปร (**variable**) เท่ากับ value1 ก็จะทำ statements1 แล้วก็ทำคำสั่ง break เพื่อออกจากคำสั่ง switch() case
- ❑ ถ้าตัวแปร (**variable**) ไม่เท่ากับ value ใดๆ เลย ก็จะไปทำ statement แล้วก็ออกจากคำสั่ง switch() case

# คำสั่ง switch() case

- สำหรับตัวแปร (**variable**) ที่ใช้กับคำสั่ง switch() case นั้น
- จะต้องเป็นตัวแปรที่มีชนิดข้อมูลเป็น **int, char** หรือ คลาส **String** เท่านั้น
- **ข้อสังเกต** หลังจากทำคำสั่งแต่ละ **case** เสร็จแล้วจะต้องปิดท้าย **statements** เหล่านั้นด้วยคำสั่ง **break** เสมอ
- ฝั่งงานของ **switch() case** จะคล้ายกับฝั่งงานคำสั่ง **nested if() else**

# ตัวอย่างการใช้งานคำสั่ง `switch()` case

□ เขียนโปรแกรมเพื่อแสดงเมนูให้ผู้ใช้เลือกดำเนินการกับตัวเลขที่รับเข้ามา 2 จำนวน โดยมีเมนู ดังนี้

กด 1 เพื่อหาผลรวมตัวเลข

กด 2 เพื่อหาผลคูณตัวเลข

กด 3 เพื่อหาค่าเฉลี่ย

กด 4 เพื่อออกจากโปรแกรม



# ตัวอย่างการใช้งานคำสั่ง switch() case

## กำหนดตัวแปร

num1	แทน ตัวเลขตัวที่1
num2	แทน ตัวเลขตัวที่2
result	แทน ผลลัพธ์
choice	แทน ตัวเลือกเมนู

# ตัวอย่างการใช้งานคำสั่ง `switch()` case

## อัลกอริทึม

1. กำหนดตัวแปร `num1, num2, result, choice`
2. รับค่า `num1` และ `num2` จากแป้นพิมพ์
3. พิมพ์ข้อความแสดงเมนูให้เลือก
4. รับค่า `choice`

# ตัวอย่างการใช้งานคำสั่ง `switch() case`

## 5. ตรวจสอบค่า `choice`

5.1 ถ้า `choice=1` ให้ `result = num1+num2` แล้วสั่งพิมพ์ `result` แล้วไปทำข้อ 6.

5.2 ถ้า `choice=2` ให้ `result = num1*num2` แล้วสั่งพิมพ์ `result` แล้วไปทำข้อ 6.

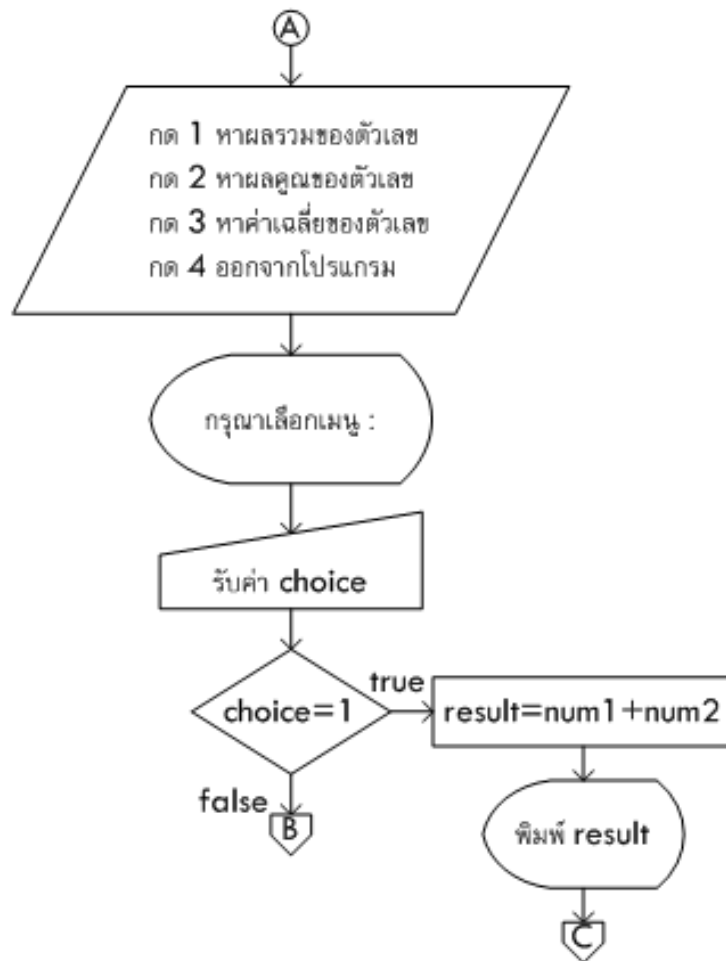
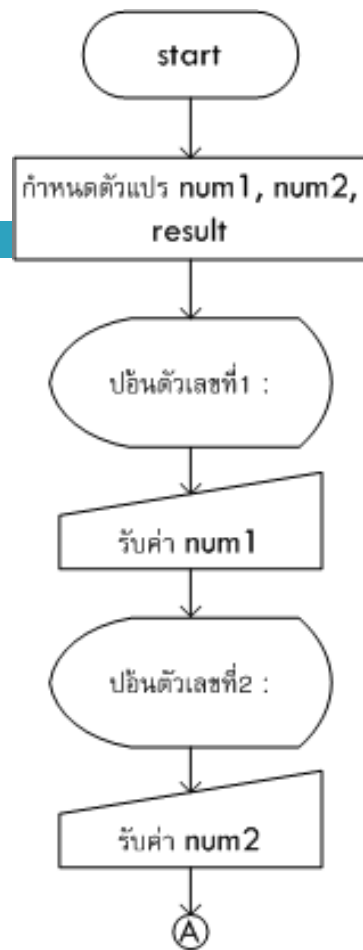
5.3 ถ้า `choice=3` ให้ `result = (num1+num2)/2` แล้วสั่งพิมพ์ `result` แล้วไปทำข้อ 6.

5.4 ถ้า `choice=4` สั่งพิมพ์ “ออกจากโปรแกรม” แล้วให้ไปทำข้อ 6.

## 6. จบการทำงาน

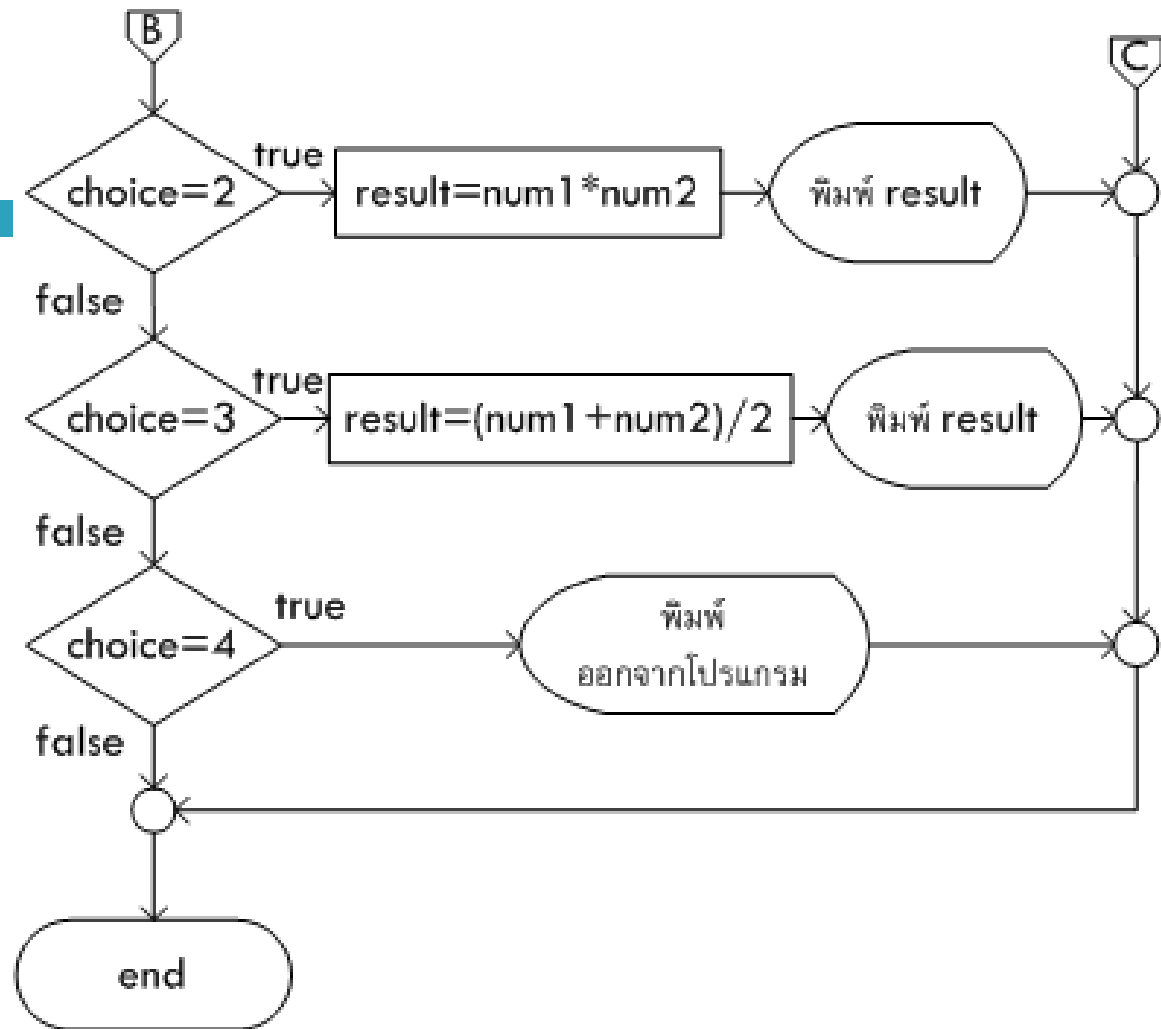
# ตัวอย่าง

□ ผังงาน



# ตัวอย่าง

□ ผังงาน



# ตัวอย่าง

☐ โค้ด

```
Menu.java ✕
1 import java.util.Scanner;
2 public class Menu {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         double num1,num2,result;
6         int choice;
7         System.out.print("ป้อนตัวเลขที่1 : ");
8         num1 = sc.nextDouble();
9         System.out.print("ป้อนตัวเลขที่2 : ");
10        num2 = sc.nextDouble();
11        System.out.println("====เมนูหลัก====");
12        System.out.println("กด 1 หาผลรวมของตัวเลข");
13        System.out.println("กด 2 หาผลคูณของตัวเลข");
14        System.out.println("กด 3 หาค่าเฉลี่ยของตัวเลข");
15        System.out.println("กด 4 ออกจากโปรแกรม");
16        System.out.println("====");
17        System.out.print("กรุณาเลือกเมนู : ");
18        choice = sc.nextInt();
```

# ตัวอย่าง

□ โค้ด

```
19      switch(choice){
20      case 1: result = num1+num2;
21              System.out.println("ผลรวม : " + result);
22              break;
23      case 2: result = num1*num2;
24              System.out.println("ผลคูณ : " + result);
25              break;
26      case 3: result = (num1+num2)/2;
27              System.out.println("ค่าเฉลี่ย : " + result);
28              break;
29      case 4: System.out.println("ออกจากโปรแกรม");
30              break;
31      }
32      sc.close();
33  }
34 }
35
```

# ตัวอย่างการใช้งานคำสั่ง switch() case

## ❑ ผลลัพธ์

```
@ Javadoc Problems Declaration Console
<terminated> Menu [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.e
ป้อนตัวเลขที่1 : 23
ป้อนตัวเลขที่2 : 67
=====เมนูหลัก=====
กด 1 หาผลรวมของตัวเลข
กด 2 หาผลคูณของตัวเลข
กด 3 หาค่าเฉลี่ยของตัวเลข
กด 4 ออกจากโปรแกรม
=====
กรุณาเลือกเมนู : 3
ค่าเฉลี่ย : 45.0
```

```
@ Javadoc Problems Declaration Console
<terminated> Menu [Java Application] C:\Program Files (x86)\Java\jre7\bin\j
ป้อนตัวเลขที่1 : 34
ป้อนตัวเลขที่2 : 12
=====เมนูหลัก=====
กด 1 หาผลรวมของตัวเลข
กด 2 หาผลคูณของตัวเลข
กด 3 หาค่าเฉลี่ยของตัวเลข
กด 4 ออกจากโปรแกรม
=====
กรุณาเลือกเมนู : 4
ออกจากโปรแกรม
```



# condition operator

- ❑ ตัวดำเนินการ **condition operator** หรือ **arithmetic if** หรือ **ternary operator** มีรูปแบบคำสั่งดังนี้

```
variable = (condition) ? statement1 : statement2;
```

- ❑ การทำงานของคำสั่ง **condition operator** จะตรวจสอบ **condition** ก่อน ถ้า **condition** เป็นจริง **variable** จะมีค่าเท่ากับ **statement1** แต่ถ้า **condition** เป็นเท็จ **variable** จะมีค่าเท่ากับ **statement2**

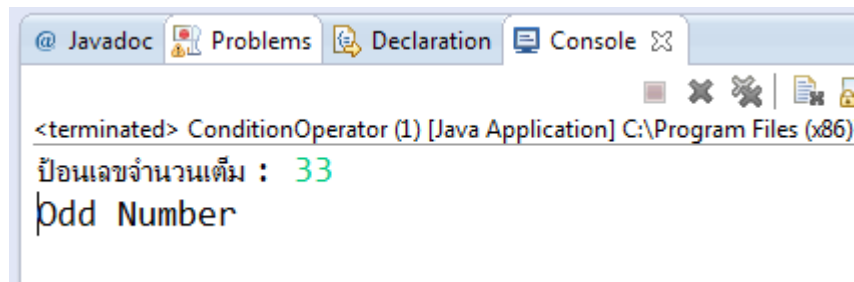
# ตัวอย่างการใช้งาน condition operator

```
Menu.java ConditionOperator.java ✕
1 import java.util.Scanner;
2 public class ConditionOperator {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         String strResult;
6         int num,result;
7         System.out.print("ป้อนเลขจำนวนเต็ม : ");
8         num = sc.nextInt();
9         result = num%2;
10        strResult = (result==0) ? "Even Number" : "Odd Number" ;
11        System.out.println(strResult);
12        sc.close();
13    }
14 }
15 |
```

ปรับปรุงจากโปรแกรมหาเลขคู่เลขคี่

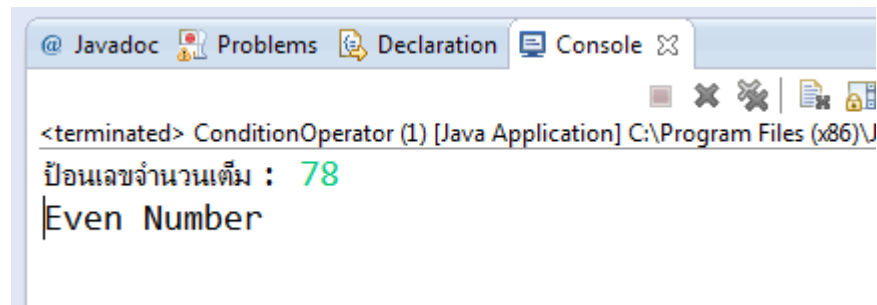
# ตัวอย่างการใช้งาน condition operator

□ ผลลัพธ์



The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar includes '@ Javadoc', 'Problems', 'Declaration', and 'Console'. The console output displays the text '<terminated> ConditionOperator (1) [Java Application] C:\Program Files (x86)\', followed by 'ป้อนเลขจำนวนเต็ม : 33' where '33' is green, and 'Odd Number' on the next line.

```
<terminated> ConditionOperator (1) [Java Application] C:\Program Files (x86)\
ป้อนเลขจำนวนเต็ม : 33
Odd Number
```



The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar includes '@ Javadoc', 'Problems', 'Declaration', and 'Console'. The console output displays the text '<terminated> ConditionOperator (1) [Java Application] C:\Program Files (x86)\', followed by 'ป้อนเลขจำนวนเต็ม : 78' where '78' is green, and 'Even Number' on the next line.

```
<terminated> ConditionOperator (1) [Java Application] C:\Program Files (x86)\
ป้อนเลขจำนวนเต็ม : 78
Even Number
```

# เอกสารอ้างอิง

พนิดา พาณิชกุล. การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา **Java**.

กรุงเทพฯ : เค ที พี คอมพ์ แอนด์ คอนซัลท์, **2548**.

สมชาย ประสิทธิ์จตุระกุล. เริ่มเรียนเขียนโปรแกรม ฉบับวาจาจาวา. กรุงเทพฯ :

สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, **2552**.

End.

