

# บทที่ 2

## ทบทวนภาษาจาวา

## Java Fundamental Review

# หลักการเขียนโปรแกรมเชิงวัตถุ

## เนื้อหา

- Java Syntax พื้นฐาน
- ตัวแปรและชนิดข้อมูล
- Operators
- Control Structures
- Arrays
- Methods
- Input/Output พื้นฐาน

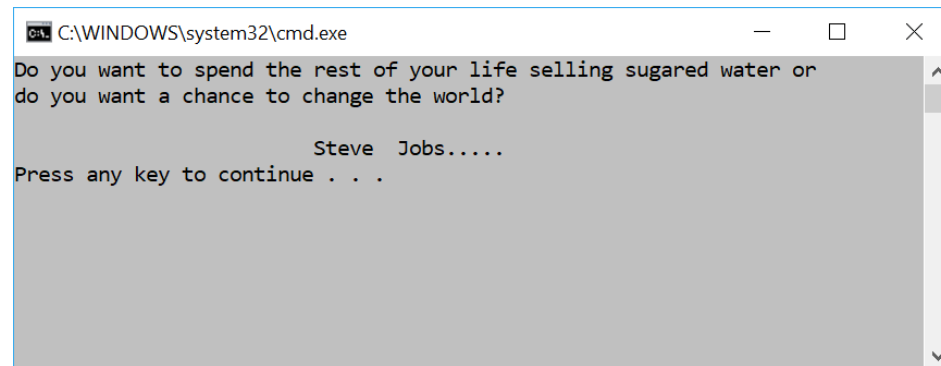
# A First Program

## ตัวอย่างที่ 1.1 DisplayMessage.java

```
/*Author : Rewadee Piputsoongnern
Purpose : display any text in a console window */

public class DisplayMessage {
    //method main() : application entry point
    public static void main(String args[]) {
        System.out.print("Do you want to spend the rest of your life ");
        System.out.println("selling sugared water or ");
        System.out.println("do you want a chance to change the world?");
        System.out.println();
        System.out.println("                                Steve Jobs.....");
    }
}
```

ผลลัพธ์

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window displays the output of the Java program: "Do you want to spend the rest of your life selling sugared water or", "do you want a chance to change the world?", followed by a blank line and "Steve Jobs.....". At the bottom, it says "Press any key to continue . . .".

```
C:\WINDOWS\system32\cmd.exe
Do you want to spend the rest of your life selling sugared water or
do you want a chance to change the world?

                                Steve Jobs.....

Press any key to continue . . .
```

# Comment (หมายเหตุ)

- Comment คือข้อความที่แทรกอยู่ในโปรแกรม มีไว้เพื่ออธิบายโปรแกรมให้ผู้อ่านเข้าใจโปรแกรมง่ายขึ้น โดยที่คอมไพเลอร์จะไม่แปลข้อความนี้ให้เป็นส่วนหนึ่งของโปรแกรม มีรูปแบบการเขียน ดังนี้

```
/*Author : Rewadee Piputsoongnern  
Purpose : display any text in a console window */
```

เป็นหมายเหตุสำหรับข้อความหลายบรรทัด โดยใช้เครื่องหมาย /\* หน้าข้อความบรรทัดแรก และใช้เครื่องหมาย \*/ ปิดท้ายข้อความสุดท้าย

```
//method main() : application entry point
```

เป็นคอมเมนต์สำหรับข้อความบรรทัดเดียว โดยใช้เครื่องหมาย // หน้าข้อความที่ต้องการเขียน จะอยู่ส่วนไหนของโปรแกรมก็ได้

# Whitespace (ช่องว่าง)

Whitespace เป็นส่วนที่ใช้เพื่อแยกคำ ประโยค หรือคำสั่งต่าง ๆ ภายในโปรแกรม โดยช่องว่างจะมีขนาดเท่าใดก็ได้ คอมไพเลอร์จะไม่นำส่วนที่เป็นช่องว่างมาเกี่ยวข้องกับขนาดของโปรแกรมไบท์โค้ด โดยช่องว่างจะทำให้รูปแบบของโปรแกรมดูง่ายขึ้น รูปแบบของช่องว่างประกอบด้วย

- **ช่องว่าง** – กดคีย์ space บนคีย์บอร์ด
- **แท็บ** – กดคีย์ tab บนคีย์บอร์ด
- **การขึ้นบรรทัดใหม่** – กดคีย์ enter บนคีย์บอร์ด

# Keyword และ Identifier

```
public class DisplayMessage {
```

keyword

keyword

identifier

วงเล็บเปิด

# Keyword (คำสงวน / คำเฉพาะ)

- **Keywords** คือชื่อที่มีความหมายพิเศษ โดยคอมไพเลอร์จะเข้าใจความหมายและคำสั่งที่จะต้องดำเนินการสำหรับคีย์เวิร์ดแต่ละตัว โดย keywords เหล่านี้ไม่สามารถนำมาตั้งเป็น Identifier ได้

abstract	boolean	break	byte	byvalue	case
cast	catch	char	c	const	continue
default	do	double	else	extends	final
finally	float	for	future	generic	goto
if	implements	import	inner	instanceof	int
interface	long	native	new	operator	outer
package	private	protected	public	rest	return
short	static	super	switch	synchronized	this
throw	throws	transient	try	var	void
volatile	while				

# Identifier (ชื่อ)

Identifier คือชื่อที่ตั้งขึ้นในภาษาจาวา ซึ่งอาจเป็นชื่อของคลาส ชื่อของตัวแปร หรือชื่อของเมธอด

- **ชื่อของคลาส** – ควรเป็นคำนาม ขึ้นต้นด้วยตัวอักษรตัวพิมพ์ใหญ่แล้วตามด้วยตัวอักษรตัวพิมพ์เล็กหรือตัวเลข โดยจะใช้ตัวอักษรพิมพ์ใหญ่เฉพาะอักษรนำของแต่ละคำที่ตามมาในชื่อ เช่น *Book, Student1, HelloWorld, GraduateStudent* เป็นต้น
- **ชื่อของตัวแปร** – ควรเป็นคำนาม ขึ้นต้นด้วยตัวอักษรพิมพ์เล็กแล้วตามด้วยตัวอักษรตัวพิมพ์เล็กหรือตัวเลข โดยอาจจะใช้ตัวอักษรพิมพ์ใหญ่ตามสำหรับอักษรนำของแต่ละคำที่ตามมาในชื่อ เช่น *x, id, name, student1, bookTitle, maxValue* เป็นต้น



# Identifier (ชื่อ)

- **ชื่อของเมธอด** – จะใช้หลักการเดียวกับการตั้งชื่อตัวแปร แต่ควรเป็นคำกริยา เช่น *getName, setName, showDetails, compute1, compute2* เป็นต้น
- **ชื่อของค่าคงที่**– ควรเป็นคำนาม ใช้อักษรตัวพิมพ์ใหญ่ทั้งหมด และจะแยกคำโดยใช้เครื่องหมาย `_` เช่น *MINIMUM, MIN\_GPA* เป็นต้น

# Separator (สัญลักษณ์แยกคำ)

- Separator เป็นสัญลักษณ์ที่ใช้ในการแยกคำที่มีความหมายแตกต่างกันออกไป

สัญลักษณ์ แยกคำ	หน้าที่
;	เพื่อระบุการสิ้นสุดของคำสั่งต่างๆ ภายในภาษาจาวา
( )	สำหรับต่อท้ายเมธอดหรือคำสั่งอื่นๆ ในภาษาจาวา เช่น if, for เป็นต้น
,	สำหรับแยกตัวแปรหรือคำสั่งในภาษาจาวา
.	เพื่อใช้ในการระบุคุณลักษณะหรือเมธอดของวัตถุ หรือใช้ในการระบุแพ็คเกจของภาษาจาวา
{ }	เพื่อระบุบล็อกคำสั่งของภาษาจาวา เช่น คำสั่งที่อยู่ภายในคลาส คำสั่งที่อยู่ภายในเมธอด หรืออยู่ภายในชุดคำสั่งโครงสร้างควบคุมต่างๆ เช่น if, while หรือ for เป็นต้น

# Methods **main** (เมธอด main)

- เมธอด **main()** เป็นเสมือนทางเข้าของแอปพลิเคชัน หากในคลาสที่สร้างขึ้นไม่มีเมธอด **main()** อยู่ ก็จะไม่สามารถเริ่มทำงานใด ๆ
- รูปแบบการประกาศเมธอด **main()**

```
public static void main(String args[]) {  
    //to do work  
}
```

# ชนิดข้อมูลพื้นฐาน (Primitive Data Types)

ชนิดข้อมูล	ขนาด(byte)	คำอธิบาย	ตัวอย่าง
byte	1	ใช้เก็บเลขจำนวนเต็ม ในช่วง -128 ถึง 127	75 -14
short	2	ใช้เก็บเลขจำนวนเต็ม ในช่วง -32,768 ถึง 32,767	619 -530
int	4	ใช้เก็บเลขจำนวนเต็ม ในช่วง -2,147,483,648 ถึง 2,147,483,647	29387 -86421
long	8	ใช้เก็บเลขจำนวนเต็ม ในช่วง -9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,808	13579286740 7362L
float	4	ใช้เก็บเลขที่มีทศนิยม 6-7 หลัก ในช่วง $-3.4 \times 10^{-38}$ ถึง $3.4 \times 10^{38}$	349.135 954F
double	8	ใช้เก็บเลขที่มีทศนิยม 14-15 หลัก ในช่วง $-1.7 \times 10^{-308}$ ถึง $1.7 \times 10^{308}$	87.5435678945 100D
boolean	1	ใช้เก็บค่าเชิงตรรกะ โดยค่าจริงจะเก็บ true ค่าเท็จจะเก็บ false	true false
char	2	ใช้เก็บค่าตัวอักษรเพียง 1 ตัว แบบ Unicode	'A' 'm' '0'

# ชนิดข้อมูลพื้นฐาน (Primitive Data Type)

- ข้อมูลพื้นฐานทั้ง 8 ชนิดนี้ นำไปใช้เป็นชนิดของตัวแปร หรือของข้อมูล ที่สามารถนำไปใช้ในการเขียนโปรแกรมในหลายแบบ

- การประกาศตัวแปรทั่วไป และการเรียกใช้งานตัวแปรแบบทั่วไปๆ

```
int x;
```

```
x = 5;
```

```
double y = 7.8;
```

- การประกาศ attribute ให้กับคลาส

```
private char grade;
```

- การส่งผ่านไปเป็นพารามิเตอร์ ให้กับ constructor ของคลาส

```
public Product(int id, double number) { ..... }
```

- การส่งผ่านไปเป็นพารามิเตอร์หรือส่วนส่งกลับใน method ของคลาส

```
public void setPrice(double d) { ..... }
```

```
public double getPrice() { ..... }
```

# การประกาศตัวแปร

## รูปแบบ

```
Type variableName;  
variableName = value;
```

### ตัวอย่าง

```
byte a;  
a = 1;  
int x, y, z;  
x = 111;  
y = 222;  
z = 333;  
char g;  
g = 'A';  
boolean h;  
h = true;
```

หรือ

```
Type variableName = value;
```

### ตัวอย่าง

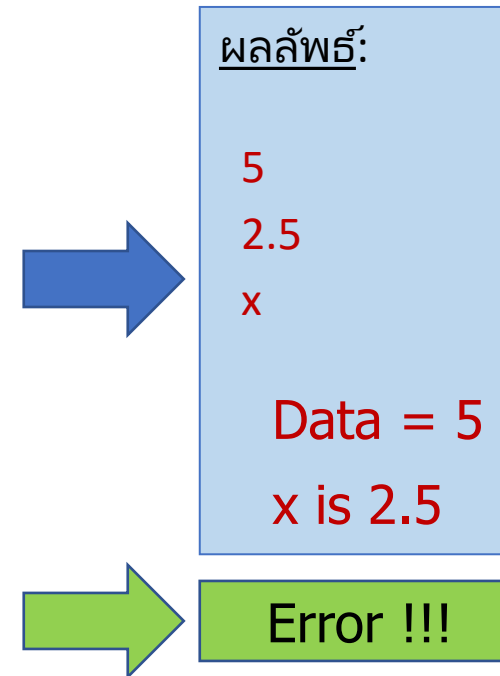
```
byte a = 1;  
short b = 222;  
int c = 12344;  
long d = 343434343;  
double e = 2.3;  
float f = 2.3f;  
char g = 'A';  
boolean h = true;  
int n1 = 10 , n2 = 20;
```

# การประกาศตัวแปร

ตัวอย่าง

```
public class Test1
{
    public static void main(String[] args)
    {
        int a = 5;
        double b = 2.5;
        char c = 'x';
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);

        System.out.println("Data = " + a);
        System.out.println(c + " is " + b);
        int d;
        System.out.println(d);
    }
}
```



ในเมธอด main() นี้ ตัวแปร d ยังไม่ได้กำหนดค่าให้กับตัวแปร จึงไม่สามารถเรียกใช้งานได้

# การประกาศตัวแปร

ตัวอย่าง

```
public class Test2
{
    public static void main(String[] args)
    {
        boolean x = true;
        int y = 10;
        double z;
        z = 3.14;
        System.out.println(x);
        System.out.println(y);
        System.out.println(z );
        System.out.println(X+ " is false");
    }
}
```



ในเมธอด **main()** นี้ไม่เคยประกาศตัวแปร **X** มาก่อน จึงไม่สามารถเรียกใช้งานได้



# การประกาศตัวแปร

## ตัวอย่าง

```
public class Test3
{
    public static void main(String[] args)
    {
        int a = 3.0;
        double b = 10;
        char c = 5;
        boolean d = 'true';
        float e = 2.3;
        double f = 13.75;
        int g = f;
    }
}
```

### การเขียนที่ถูกต้อง



Error !!!

int a = 3;



Error !!!

char c = '5';



Error !!!

boolean d = true;



Error !!!

float e = 2.3f;



Error !!!

int g = (int)f;

# การประกาศตัวแปร

- ไม่ควรประกาศค่าให้กับตัวแปร เกินช่วงของข้อมูล เช่น

```
int n = 1000000;  
System.out.println(n * n); // prints -727379968
```

- การแปลงระหว่างตัวเลขทศนิยมให้เป็นจำนวนเต็ม อาจมีข้อผิดพลาด

```
double f = 4.35;  
System.out.println(100 * f); // prints 434.99999999999994
```

- การกำหนดค่าทศนิยม ให้กับเลขจำนวนเต็ม จะเกิดข้อผิดพลาดขึ้น

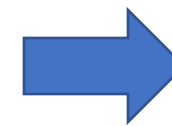
```
double balance = 13.75;  
int dollars = balance; // Error
```

- ในการเรียกใช้ตัวแปรใน method ใดๆ จะสามารถเรียกใช้ได้ ก็ต่อเมื่อ เป็นตัวแปร local ใน method นั้น

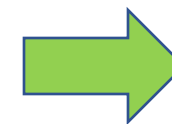
# การประกาศตัวแปร

ตัวอย่าง

```
public class Test4
{
    private double data;
    public int compute()
    {
        int a = 5;
        return a;
    }
    public static void main(String[] args)
    {
        boolean x=false;
        int y = 10;
        System.out.println(x);
        System.out.println("Value =" +y
    );
        System.out.println(data);
        System.out.println(a);
    }
}
```



ผลลัพธ์:  
false  
Value=10



Error !!!

Error !!!

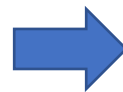
ตัวแปร data และตัวแปร a อยู่นอกเมธอด main() ไม่ใช่ตัวแปร local จึงไม่สามารถเรียกใช้งานได้

# การประกาศตัวแปร

## ตัวอย่าง

```
public class Test5
{
    private double data;
    public Test5()
    {
        data = 0.5;
    }
    public Test5(double d)
    {
        data = d;
    }
    public double getData()
    {
        return data;
    }
    public static void main(String[] args)
    {
        double x = 1.0;
        Test5 t1 = new Test5();
        Test5 t2 = new Test5(x);
        System.out.println(t1.getData() + x);
        System.out.println(t2.getData() );
        System.out.println(data);
    }
}
```

ตัวแปร data เป็นแอตทริบิวต์ของคลาสที่อยู่นอก main() จึงไม่สามารถเรียกใช้งานได้โดยตรงถ้าจะเรียกใช้ค่าของแอตทริบิวต์ data ต้องเรียกว่าเมธอด getData()



ผลลัพธ์:

1.5

1.0

Error !!!

# การประกาศตัวแปร

## Local vs Instance Variable

- Local : ประกาศภายใน method
- Instance : ประกาศภายในคลาสอยู่นอก method

# ค่าคงที่ (Constants)

- การประกาศตัวแปรค่าคงที่จะใช้คำว่า **final** นำหน้า
- เมื่อกำหนดค่าให้กับค่าคงที่แล้ว ไม่สามารถเปลี่ยนแปลงค่าได้
- ชื่อของตัวแปรค่าคงที่ต้องเขียนให้สามารถเข้าใจได้ง่าย โดยต้องตั้งชื่อเป็นอักษรตัวใหญ่ทั้งหมด  
เช่น ตัวแปรชื่อ PI, MAX, FIX เป็นต้น

# ค่าคงที่ (Constants)

ในเมธอดทั่วไป รวมทั้งเมธอด main:

```
final typeName variableName = value;
```

ในคลาส:

```
modifier static final typeName variableName = value;
```

ตัวอย่าง:

ในเมธอด

```
final double NICKEL_VALUE = 0.05;
```

ในคลาส

```
public static final double LITERS_PER_GALLON = 3.785;
```

# ค่าคงที่ (Constants)

## ตัวอย่าง

```
public class Test6
{
    public static void main(String args[])
    {
        final double PI = 3.14;
        double radius = 2;
        System.out.println("Area = "+(PI*radius*radius));
        PI = PI + 1;
    }
}
```

ตัวแปร PI เป็นค่าคงที่ ไม่สามารถเปลี่ยนแปลงค่าได้



ผลลัพธ์:  
Area = 12.56

Error !!!



# ค่าคงที่ (Constants)

- ถ้าค่าคงที่จะถูกนำไปใช้ในหลายเมธอด ควรประกาศค่าคงที่นั้นให้เป็น แอททริบิวต์ของคลาส และนำหน้าด้วย **static final**
- สามารถเพิ่มการเข้าถึงด้วย **public** ก่อนหน้า **static final** เพื่อให้คลาสอื่นๆ สามารถเข้าถึงค่าคงที่เหล่านั้น

```
public class MyClass
{   public static final double E = 2.7182818284590452354;
    public static final double PI = 3.14159265358979323846;
    . . .
}
```

- การเรียกใช้งานค่าคงที่ของคลาสในเมธอดต่าง ๆ สามารถเรียกโดยขึ้นต้นด้วยชื่อคลาสตามด้วย . ตามด้วยชื่อตัวแปรค่าคงที่โดยตรง เช่น

```
System.out.println(MyClass.PI);
```

# ค่าคงที่ (Constants)

## ตัวอย่าง

```
public class Test7 {  
    public double data = 2.72;  
    public static final double PI = 3.14592;  
    public static void main(String[] args) {  
        final double PI = 3.14;  
        double radius = 2;  
        System.out.println("Circle Area = " + (PI*radius*radius));  
        System.out.println("Circle Area = " + (Test7.PI*radius*radius));  
        System.out.println("Test Area = " + (data*radius*radius));  
    }  
}
```

ผลลัพธ์:

Circle Area = 12.56

Circle Area = 12.58363

Error !!!

# String และการจัดการข้อความ

- String Basics

```
String greeting = "Hello World";  
String name = new String("Java");
```

```
// String Methods
```

```
int length = greeting.length();           // 11  
char firstChar = greeting.charAt(0);       // 'H'  
String upper = greeting.toUpperCase();     // "HELLO WORLD"  
String sub = greeting.substring(0, 5);    // "Hello"  
boolean contains = greeting.contains("World"); // true
```

length() : หาขนาดของ String

charAt() : ตัวอักขระในตำแหน่งที่ระบุ

toUpperCase() : ปรับข้อความให้อักษรตัวพิมพ์ใหญ่

toLowerCase() : ปรับข้อความให้เป็นตัวพิมพ์เล็ก

substring() : ตัดเอาข้อความตามตำแหน่งที่ระบุ

contains() : ตรวจสอบข้อความใน String

# String และการจัดการข้อความ

- String Concatenation(การต่อ/รวมข้อความ)

```
String firstName = "John";  
String lastName = "Doe";  
String fullName = firstName + " " + lastName;  
// หรือ  
String fullName2 = firstName.concat(" ").concat(lastName);
```

# นิพจน์ทางคณิตศาสตร์ (Arithmetic operator)

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operator)

Operator	Java operator	Example	Value (x = 10 , y = 7 , z = 2.5)
บวก (Addition)	+	x + y	17
ลบ (Subtration)	-	x - y	3
คูณ (Multiplication)	*	x * y	70
หาร (Division)	/	x / y x / z	1 4.0
หารเอาเศษ (Modulo division)	%	x % y	3
บวกเพิ่มค่าทีละ1	++	x++ หรือ ++x	11
ลดค่าลงทีละ 1	--	x-- หรือ --x	9

# นิพจน์ทางคณิตศาสตร์ (Arithmetic operator)

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operator)

Operator	Java operator	Example	Value (x = 10 , y = 7 , z = 2.5)
บวกลดรูป	+=	x += y	x = 17
ลบลดรูป	-=	x -= y	x = 3
คูณลดรูป	*=	x *= y	x = 70
หารลดรูป	/=	x /= y z /= 2	x = 1 z = 1.0
หารเอาเศษลดรูป	%=	x %= y	x = 3

# ลำดับความสำคัญ (Precedence)

Precedence	Operator	Type	Associativity
 <div>ก่อน</div>	() [] .	Parentheses Array subscript Member selection	Left to Right
	++ --	Unary post-increment Unary post-decrement	Right to left
	++ -- + - ! ~ ( type )	Unary pre-increment Unary pre-decrement Unary plus Unary minus Unary logical negation Unary bitwise complement Unary type cast	Right to left
	* / %	Multiplication Division Modulus	Left to right
	+ -	Addition Subtraction	Left to right
	<< >> >>>	Bitwise left shift Bitwise right shift with sign extension Bitwise right shift with zero extension	Left to right
	< <= > >= instanceof	Relational less than Relational less than or equal Relational greater than Relational greater than or equal Type comparison (objects only)	Left to right
	== !=	Relational is equal to Relational is not equal to	Left to right
	&	Bitwise AND	Left to right
	^	Bitwise exclusive OR	Left to right
		Bitwise inclusive OR	Left to right
	&&	Logical AND	Left to right
		Logical OR	Left to right
	? :	Ternary conditional	Right to left
	= += -= *= /=	Assignment Addition assignment Subtraction assignment Multiplication assignment Division assignment	Right to left
	%=	Modulus assignment	
			หลัง

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

- ❖ การดำเนินการทางคณิตศาสตร์ จะทำจากซ้ายไปขวา
- ❖ การดำเนินการทางคณิตศาสตร์ จะทำตามระดับความสำคัญของตัวดำเนินการ เช่น  
ทำ \* / % ก่อน + -
- ❖ ( ) เป็นเครื่องหมายที่ใช้ในการแยกกลุ่ม โดยถ้าพบเครื่องหมายนี้ จะทำการประมวลผลทางคณิตศาสตร์ภายในวงเล็บนี้ก่อน
- ❖ / เป็นเครื่องหมายที่ใช้ในการหาร  
 $7.0 / 4$  ได้ผลลัพธ์เป็น 1.75  
 $7 / 4$  ได้ผลลัพธ์เป็น 1
- ❖ % เป็นเครื่องหมายที่ใช้ใน การหาผลเศษ เรียกว่า "modulo"  
 $7 \% 4$  is 3



# การดำเนินการทางคณิตศาสตร์กับตัวแปร

```
int a = 5;  
int b = 8;  
int c = a + b;  
a = 2 * b;  
b = b - a;
```



a มีค่าเป็น 5  
b มีค่าเป็น 8  
c มีค่าเป็น 13  
a มีค่าเป็น 16  
b มีค่าเป็น -8

```
int x = 6;  
int y = 6;  
int z = 6;  
x++;  
y--;  
z+=6;
```



x มีค่าเป็น 6  
y มีค่าเป็น 6  
z มีค่าเป็น 6  
x มีค่าเป็น 7  
y มีค่าเป็น 5  
z มีค่าเป็น 12

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

```
int a = 5;  
int b = 2;  
double c = 5.0;  
double d = 2.5;
```

```
int x = a/b;  
double y = a/c;  
double z = d/b;  
double aa = c/d;
```

```
int bb = a%b;  
double cc = a%b;  
double dd = d%c;
```



x มีค่าเป็น 2
y มีค่าเป็น 1.0
z มีค่าเป็น 1.25
aa มีค่าเป็น 2.0



bb มีค่าเป็น 1
cc มีค่าเป็น 1.0
dd มีค่าเป็น 0.0

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

```
int a = 5;  
int b = 2;  
int c = 4;  
double d = 4.0;
```

```
int e = a+b*c;  
double f = a-b/c+a;  
double g = a+b/d+a;  
double h = a*b-(c-1);
```



e มีค่าเป็น 13

f มีค่าเป็น 10.0

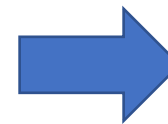
g มีค่าเป็น 10.5

h มีค่าเป็น 7.0

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

ตัวอย่าง

```
public class Test8
{
    public static void main(String[] args)
    {
        int a = 5; int b = 2;
        int c = 4; double d = 4.0;
        int e = a+b*c;
        double f = a-b/c+a;
        double g = a+b/d+a;
        double h = a*b-(c-1);
        System.out.println(e);
        System.out.println(f);
        System.out.println(g);
        System.out.println(h);
    }
}
```



ผลลัพธ์:

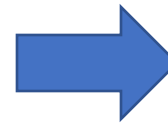
13  
10.0  
10.5  
7.0

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

## ตัวอย่าง

ให้เขียนโปรแกรม Number เพื่อหาผลรวมและค่าเฉลี่ยของเลขจำนวนเต็ม 5 ค่าต่อไปนี้ 2, 4, 6, 7, และ 1. จากนั้นแสดงผลรวมและค่าเฉลี่ย

```
public class Number
{
    public static void main(String[] args)
    {
        int x1=2, x2=4, x3=6, x4=7, x5=1;
        int sum = x1 + x2 + x3 + x4 + x5;
        double avg = sum/5;
        System.out.println("Sum = "+sum);
        System.out.println("Average = "+avg);
    }
}
```



ผลลัพธ์:

Sum = 20  
Average = 4.0

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

## ตัวอย่าง

จากโจทย์เดียวกัน ให้เขียนคลาส Number ที่ประกอบด้วย 5 แอททริบิวต์ที่ใช้เก็บเลขจำนวนเต็ม 5 ตัว สมมติมีวัตถุหนึ่งมีค่าตัวเลขทั้งห้าเป็น 2, 4, 6, 7, และ 1.... จงแสดงผลรวมและค่าเฉลี่ยของวัตถุนั้น

```
public class Number
{
    private int no1, no2, no3, no4, no5;

    public Number(int n1, int n2, int n3, int n4, int
n5)
    {
        no1 = n1;
        no2 = n2;
        no3 = n3;
        no4 = n4;
        no5 = n5;
    }

    public int findSum()
    {
        return (no1+no2+no3+no4+no5);
    }
}
```

```
public static void main(String [] args)
{
    Number n = new Number(2,4,6,7,1);
    int sum = n.findSum();
    double avg = sum/5;
    System.out.println("Sum = "+sum);
    System.out.println("Average = "+avg);
}
}
```

ผลลัพธ์:

Sum = 20  
Average = 4.0

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

## ตัวอย่าง

ให้เขียนโปรแกรม Book ที่ทำงานภายใน main() โดยให้กำหนดตัวแปรชื่อหนังสือมีค่าเป็น “Java” ตัวแปรราคามีค่าเป็น 200 เมื่อหักภาษี 7% จะได้ตัวแปรราคาสุทธิ .... จงแสดงค่าผลลัพธ์ทั้งหมดของตัวแปรเหล่านั้น

```
public class Test9
{
    public static void main(String [] args)
    {
        String name = "Java";
        double price = 200;
        double netPrice = price - (price * 7 / 100);
        System.out.println(name + " " + price+" " + netPrice);
    }
}
```

# การดำเนินการทางคณิตศาสตร์กับตัวแปร

## ตัวอย่าง

จากโจทย์เดียวกัน ให้เขียนคลาส Book ที่ประกอบด้วยแอตทริบิวต์ ชื่อหนังสือ ราคา และราคาสุทธิ โดยสมมติหนังสือชื่อ “Java” ราคาเป็น 200 บาท เมื่อหักภาษี 7% จะได้เป็นราคาสุทธิ .... จงแสดงค่าผลลัพธ์ทั้งหมดออกทางหน้าจอ

```
public class Book
{
    private String name;
    private double price;
    private double netPrice;
    public Book(String n, double p)
    {
        name = n;
        price = p;
    }
    public void setNetPrice()
    {
        netPrice = price - (price * 7 / 100);
    }
    public String getName()
    {
        return name;
    }
}
```

```
public double getPrice()
{
    return price;
}
public double getNetPrice()
{
    return netPrice;
}
public static void main(String [] args)
{
    Book b = new Book("Java", 200);
    b.setNetPrice();
    System.out.println(b.getName() + "
" + b.getPrice()+" " + b.getNetPrice());
}
```



# คำสั่งควบคุมการทำงาน

## ❖ การตัดสินใจ/เลือกทำ (Decision/Selection Statement)

- คำสั่ง if (if statement)
- คำสั่ง if...else (if...else statement)
- คำสั่ง switch (switch statement)

## ❖ การวนซ้ำ (Repetition Statement)

- คำสั่ง for (for loop statement)
- คำสั่ง while (while loop statement)
- คำสั่ง do...while (do...while statement)

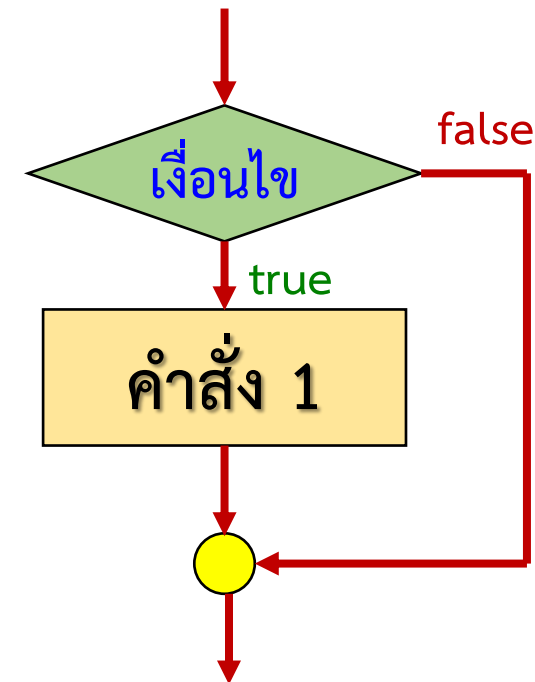
# การตัดสินใจ/เลือกทำ (Decision/Selection Statement)

# คำสั่ง if (if Statement)

**if** เป็นคำสั่งสำหรับการตัดสินใจ โดยถ้าเงื่อนไขในการตัดสินใจ ให้ผลลัพธ์เป็นจริง จะทำงานในส่วนที่กำหนดไว้ แต่ถ้าเงื่อนไขเป็นเท็จจะออกจากประโยคคำสั่งไปทำงานในคำสั่งอื่น ๆ ต่อไป

## รูปแบบคำสั่ง

```
if (เงื่อนไข)  
    คำสั่งที่จะให้ทำเมื่อเงื่อนไขมีค่าเป็น true ;
```

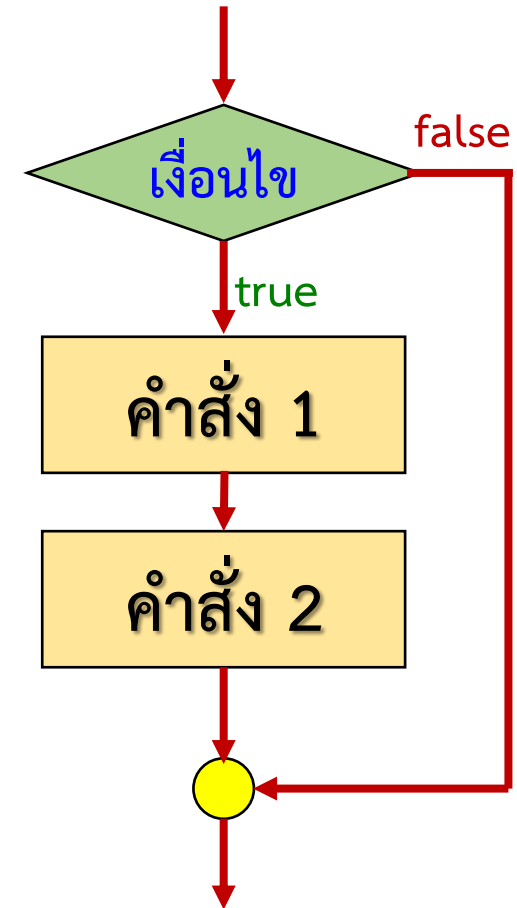


# คำสั่ง if (if Statement)

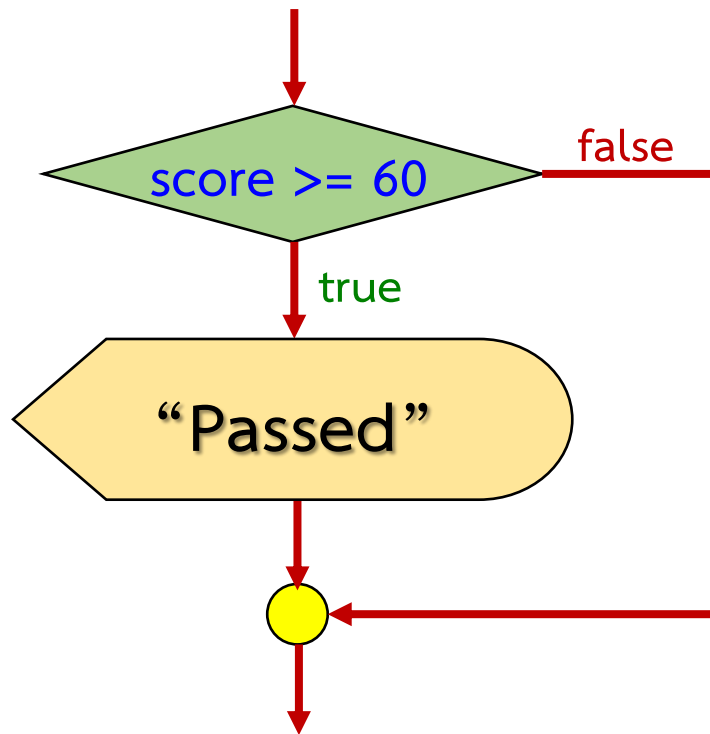
- ในกรณีมีประโยคคำสั่งที่จะต้องทำหลายคำสั่ง

## รูปแบบคำสั่ง

```
if (เงื่อนไข) {  
    คำสั่ง 1;  
    คำสั่ง 2;  
}
```



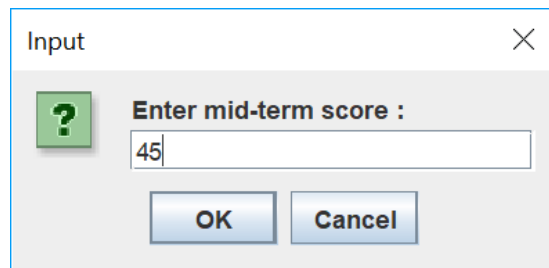
# คำสั่ง if (if Statement)



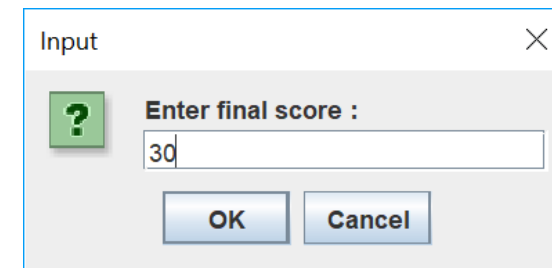
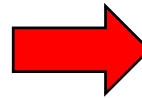
```
if(score >= 60)
    System.out.println("Passed")
```

# คำสั่ง if (if Statement)

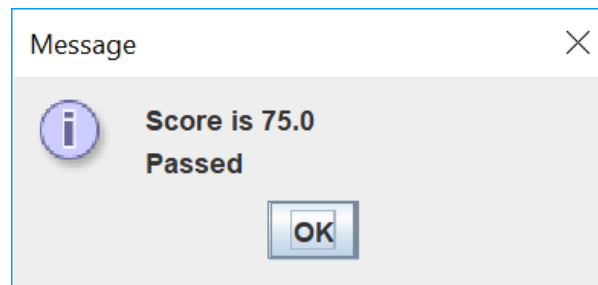
Ex1. จงเขียนโปรแกรมรับคะแนนกลางภาคและคะแนนปลายภาค คำนวณคะแนนรวม ถ้าคะแนนรวมตั้งแต่ 60 ขึ้นไป แสดงข้อความ Passed



Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Enter mid-term score :", a text input field containing "45", and two buttons: "OK" and "Cancel".



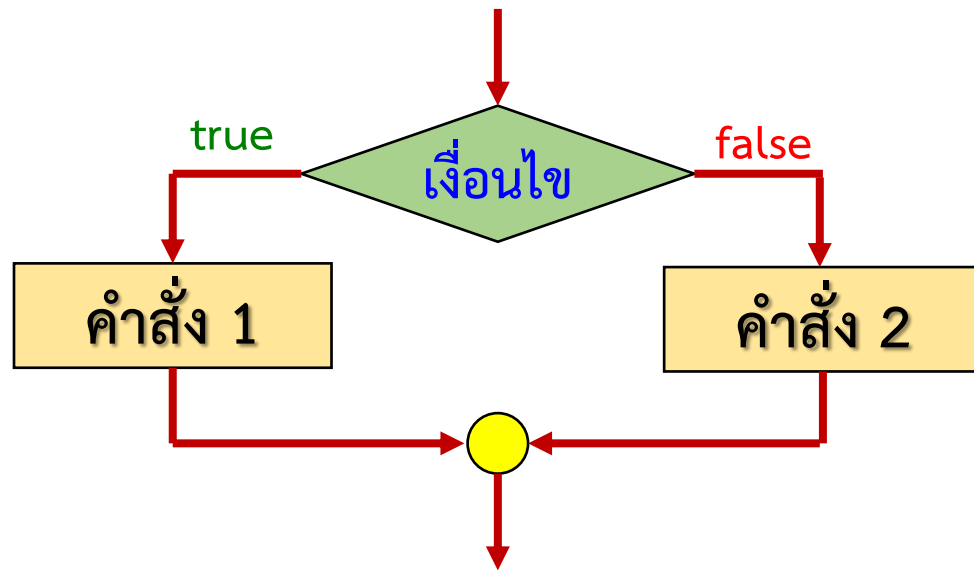
Input dialog box titled "Input" with a close button (X). It contains a green question mark icon, the text "Enter final score :", a text input field containing "30", and two buttons: "OK" and "Cancel".



Message dialog box titled "Message" with a close button (X). It contains an information icon (i), the text "Score is 75.0" and "Passed", and an "OK" button.

# คำสั่ง if..else (if..else Statement)

**if..else** เป็นคำสั่งสำหรับการตัดสินใจ โดยถ้าเงื่อนไขในการตัดสินใจ ให้ผลลัพธ์เป็นจริง จะทำงานในส่วนหลังคำสั่ง if แต่ถ้าเงื่อนไขเป็นเท็จจะทำงานในส่วนหลังคำสั่ง else

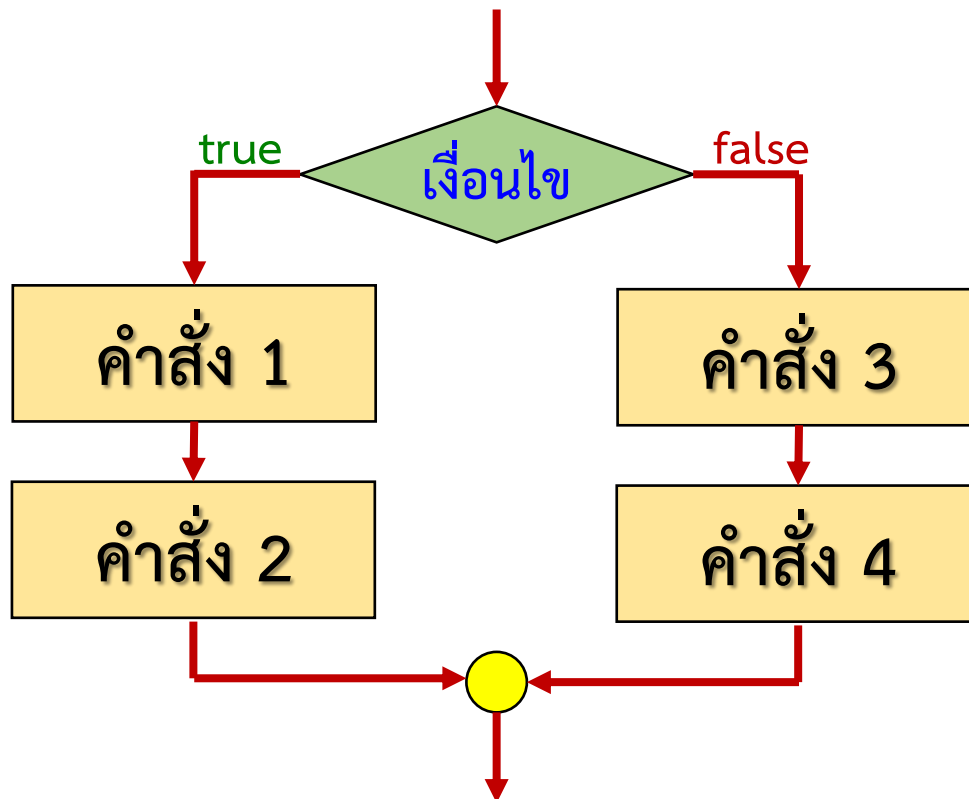


## รูปแบบคำสั่ง

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

# คำสั่ง `if..else` (`if..else` Statement)

- ในกรณีมีประโยคคำสั่งที่จะต้องทำหลายคำสั่ง

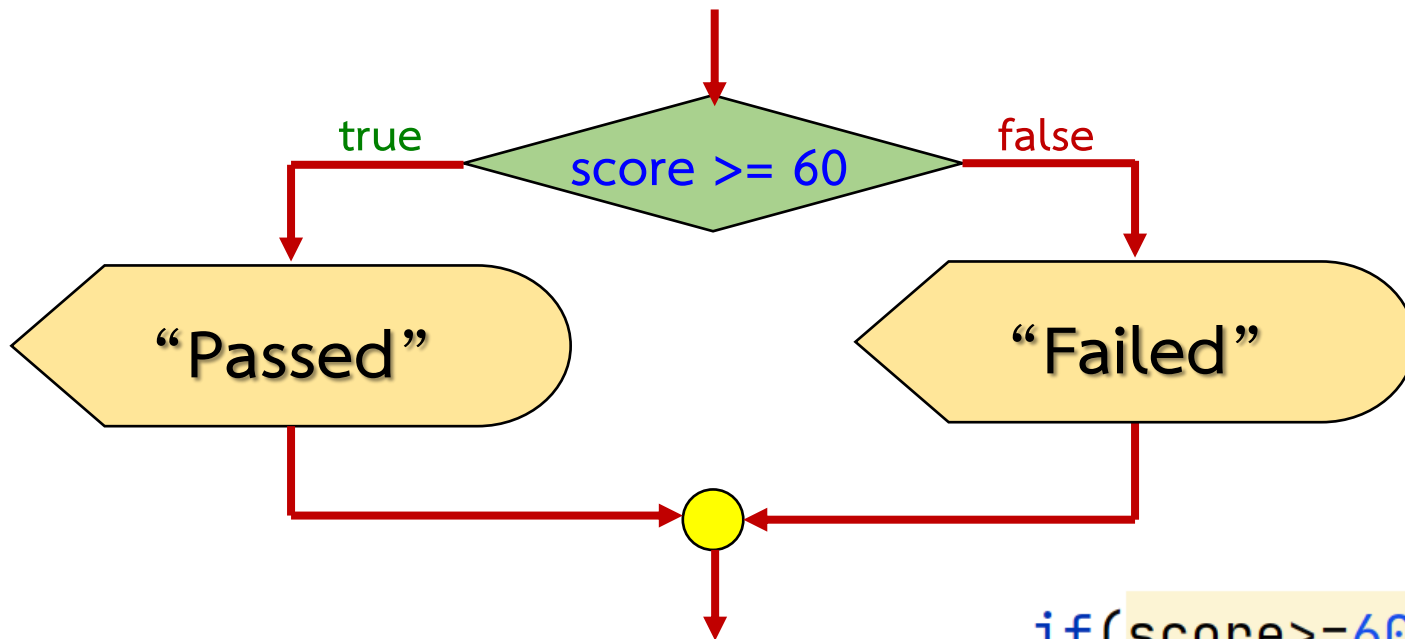


## รูปแบบคำสั่ง

```
if (เงื่อนไข) {  
    คำสั่ง 1;  
    คำสั่ง 2;  
}  
else {  
    คำสั่ง 3;  
    คำสั่ง 4;  
}
```



# คำสั่ง if..else (if..else Statement)



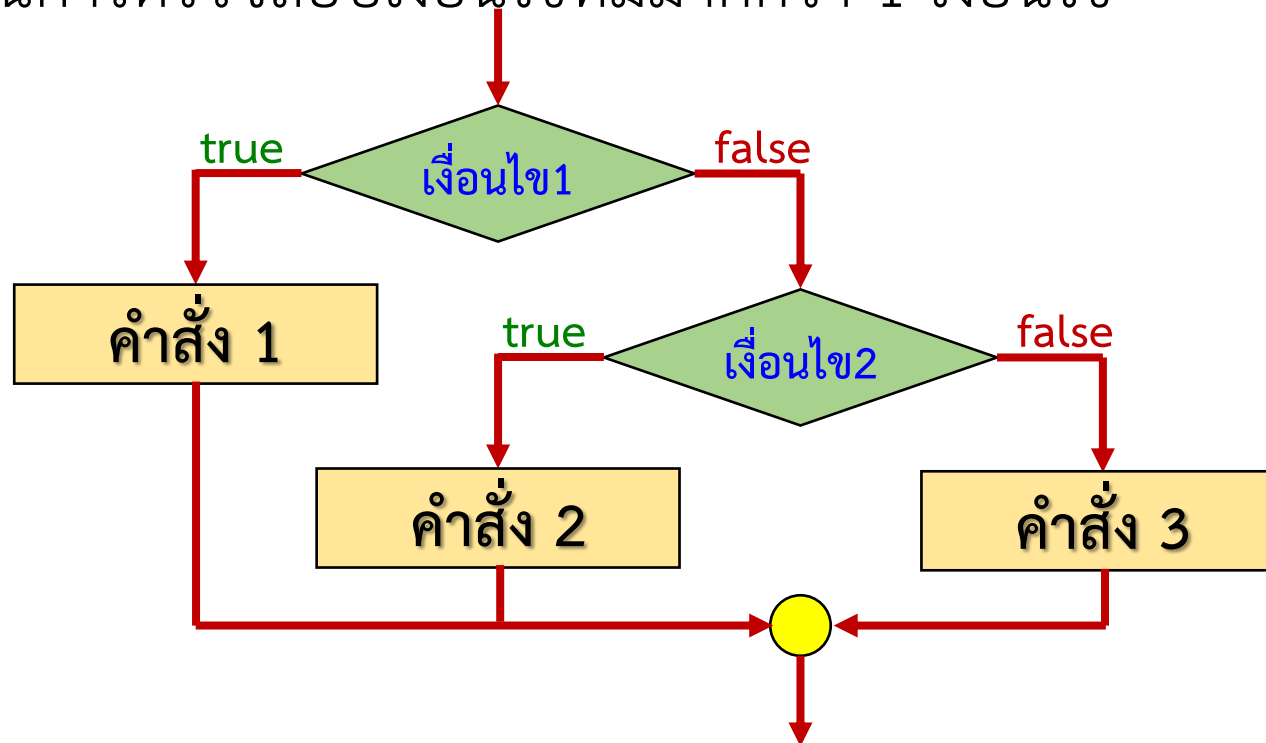
```
if(score>=60) {  
    System.out.println("Passed");  
} else {  
    System.out.println("Failed");  
}
```

# คำสั่ง if-else-if (if-else-if Statement)

- คำสั่ง if-else-if เป็นคำสั่งในการตรวจสอบเงื่อนไขที่มีมากกว่า 1 เงื่อนไข

## รูปแบบคำสั่ง

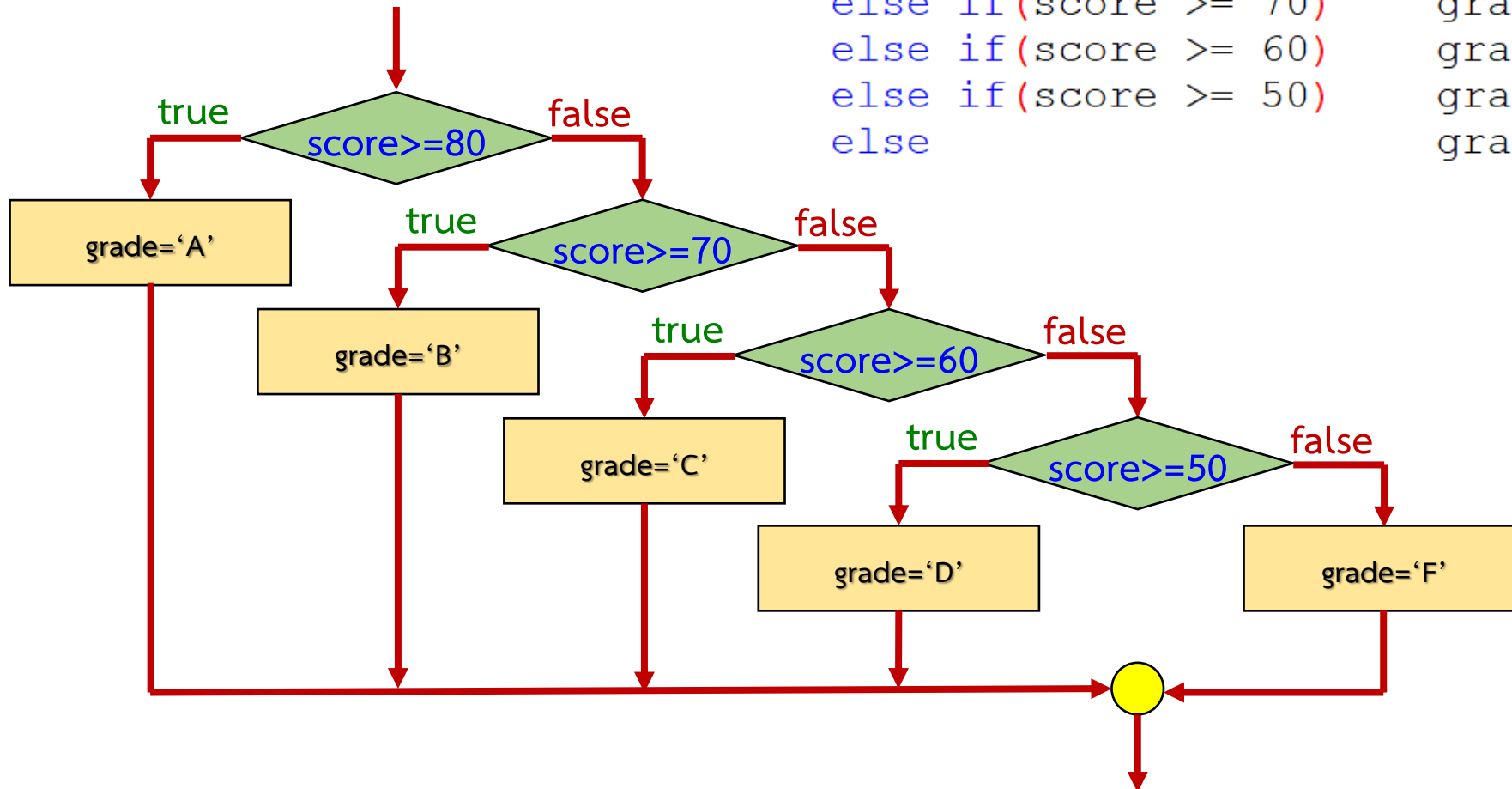
```
if (เงื่อนไข1)  
    คำสั่ง 1;  
else if (เงื่อนไข2)  
    คำสั่ง 2;  
else  
    คำสั่ง 3;
```



# คำสั่ง if-else-if (if-else-if Statement)

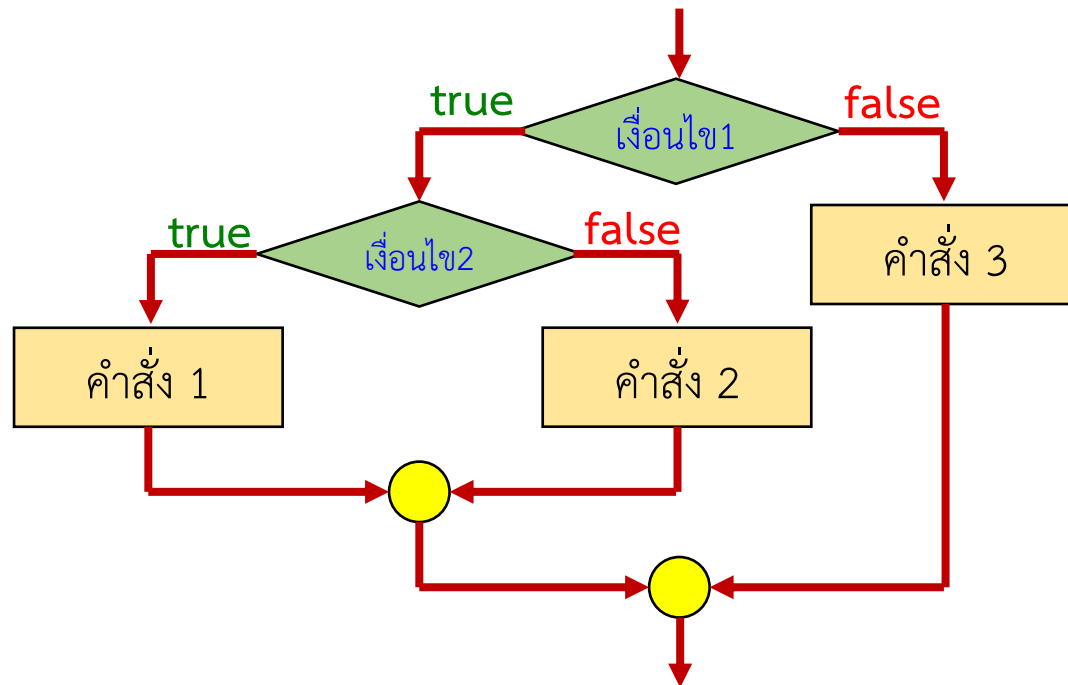
```
if(score >= 80)
else if(score >= 70)
else if(score >= 60)
else if(score >= 50)
else
```

```
grade = 'A';
grade = 'B';
grade = 'C';
grade = 'D';
grade = 'F';
```



# คำสั่ง if แบบซ้อน (Nested if)

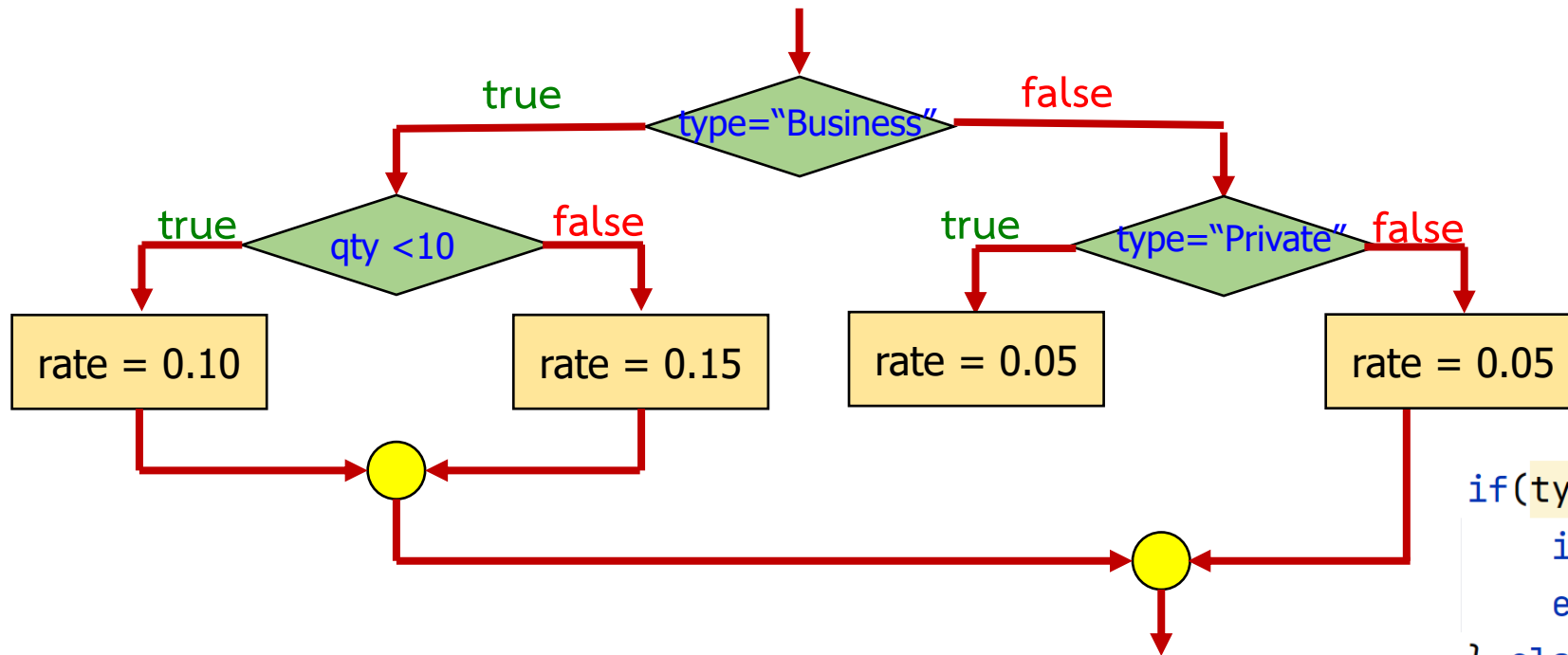
- คำสั่ง if แบบซ้อน เป็นการตรวจสอบเงื่อนไขต่าง ๆ ที่หลากหลาย โดยการซ้อนกันของ if อาจไปซ้อนเมื่อเงื่อนไขในนิพจน์เป็นจริง หรือเมื่อเงื่อนไขในนิพจน์เป็นเท็จก็ได้



## รูปแบบคำสั่ง

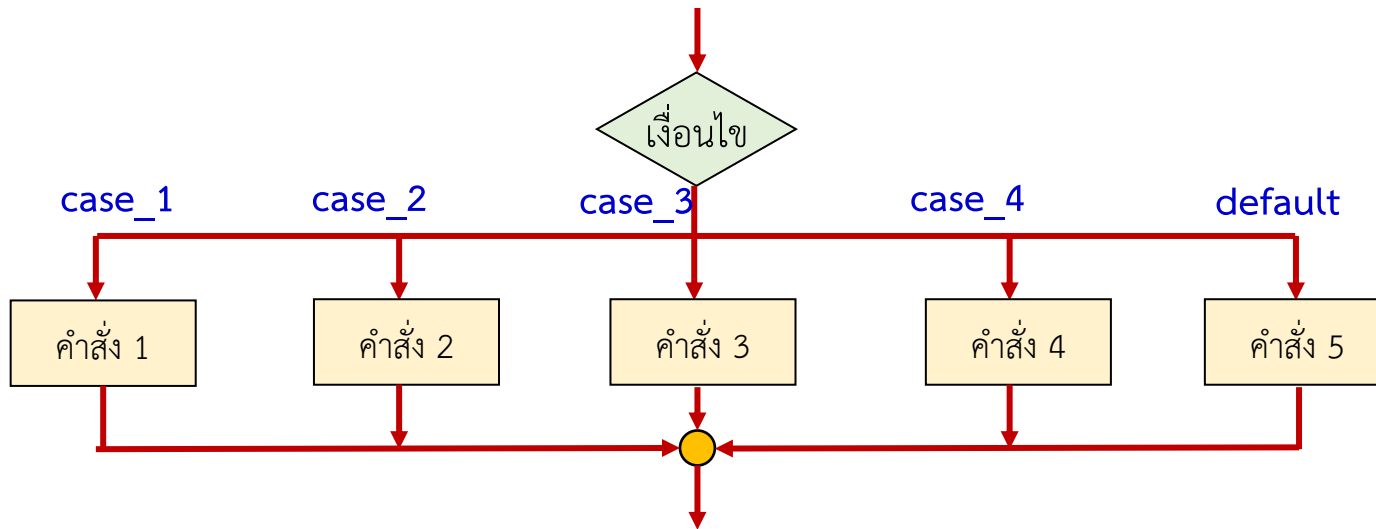
```
if (เงื่อนไข 1) {  
    if(เงื่อนไข 2) {  
        คำสั่ง 1 ;  
    }  
    else {  
        คำสั่ง 2 ;  
    }  
    else {  
        คำสั่ง 3 ;  
    }  
}
```

# คำสั่ง if แบบซ้อน (Nested if)



```
if(type.equals("Business")){  
    if(qty < 10)    rate = 0.10;  
    else           rate = 0.15;  
} else if(type.equals("Private")) {  
    rate = 0.05;  
} else {  
    rate = 0.00;  
}
```

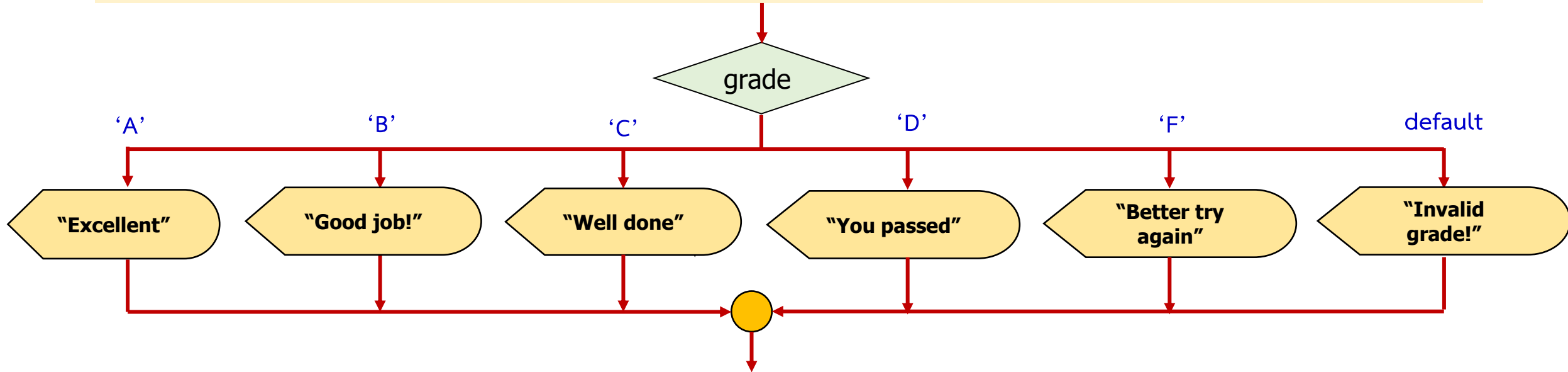
# คำสั่ง `switch...case`



## รูปแบบคำสั่ง

```
switch (เงื่อนไข) {  
    case ค่าเปรียบเทียบ1 :    คำสั่ง 1 ;  
                                break;  
    case ค่าเปรียบเทียบ2 :    คำสั่ง 2 ;  
                                break;  
    case ค่าเปรียบเทียบ3 :    คำสั่ง 3 ;  
                                break;  
    case ค่าเปรียบเทียบ4 :    คำสั่ง 4 ;  
                                break;  
    default :                  คำสั่ง 5 ;  
}
```

# คำสั่ง `switch...case`



# คำสั่ง `switch...case`

```
switch(grade) {  
    case 'A' :  
        System.out.println("Excellent!");  
        break;  
    case 'B' :  
        System.out.println("Good job!");  
        break;  
    case 'C' :  
        System.out.println("Well done");  
        break;  
    case 'D' :  
        System.out.println("You passed");  
        break;  
    case 'F' :  
        System.out.println("Better try again");  
        break;  
    default:  
        System.out.println("Invalid grade!!");  
}
```



# การวนซ้ำ (Iteration Statement)

# การวนซ้ำ (Iteration)

- โครงสร้างการเขียนโปรแกรมแบบวนซ้ำ (Iteration) หรือ Loop ใช้เพื่อให้โปรแกรมวนทำงานตามคำสั่งซ้ำๆ ตามจำนวนรอบที่ต้องการ หรือจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ จึงจะสามารถหลุดออกจากรอบการทำงานวนซ้ำนั้นได้
- การจะวนซ้ำกี่รอบนั้น ขึ้นอยู่กับเงื่อนไขที่กำหนดขึ้น บางโปรแกรมอาจกำหนดรอบการทำงานแน่นอนว่าให้วนซ้ำกี่รอบ บางโปรแกรมขึ้นอยู่กับเงื่อนไขที่ตรวจสอบ

# การวนซ้ำ (Iteration)

คำสั่งทำซ้ำในภาษา java ประกอบด้วย

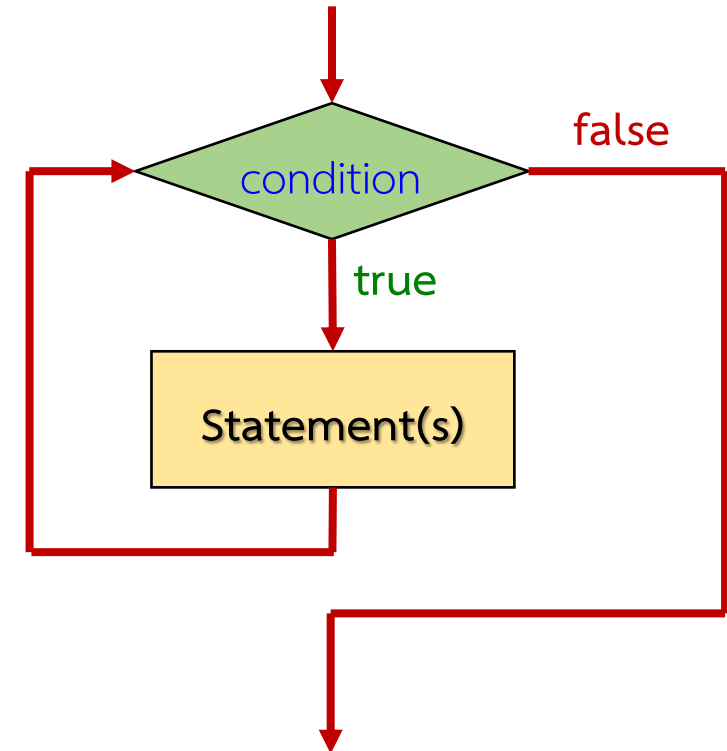
- คำสั่ง while
- คำสั่ง do...while
- คำสั่ง for

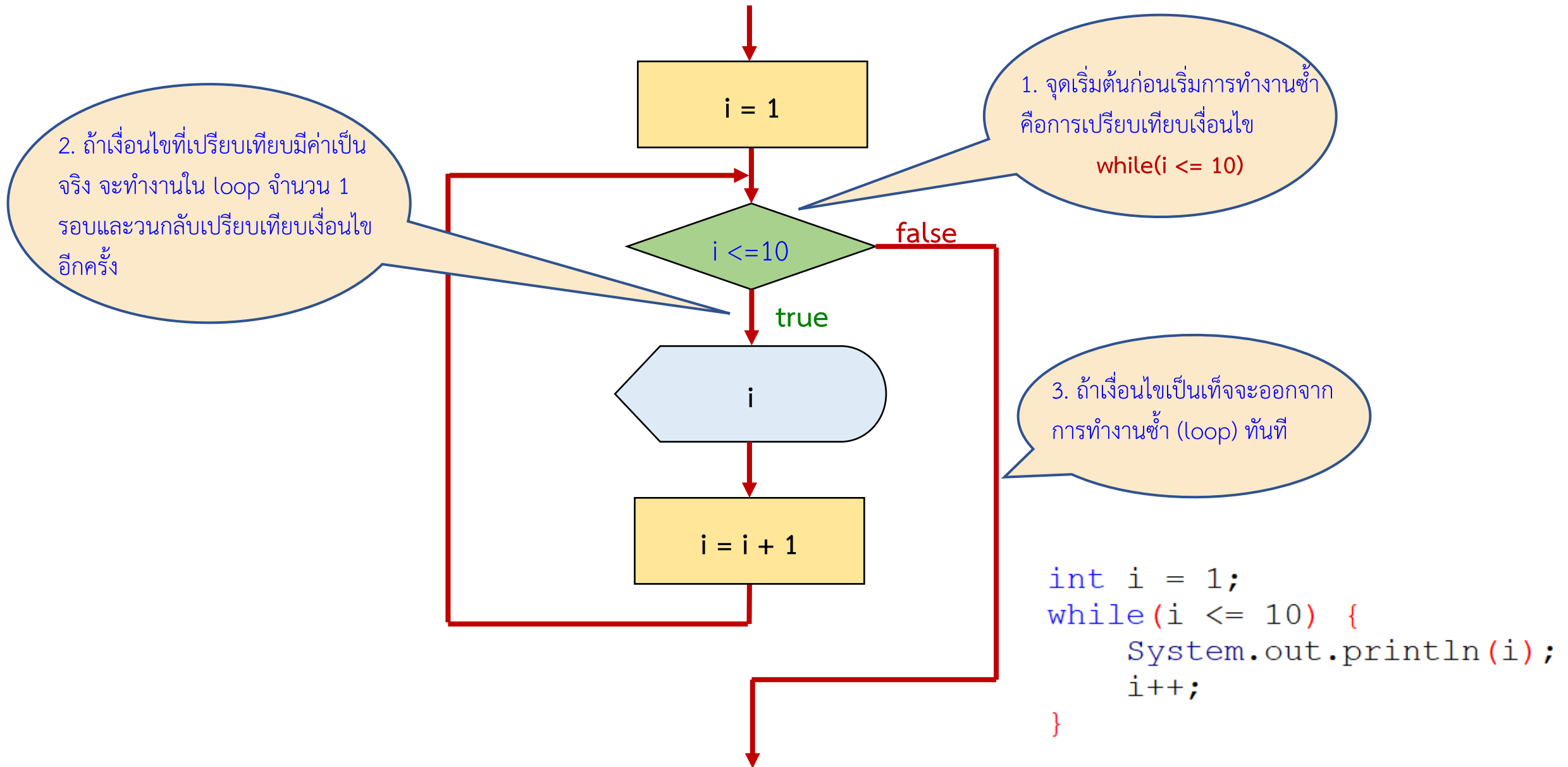
# คำสั่ง **while** (while statement)

- คำสั่ง **while** ใช้สำหรับวนซ้ำทำงานของกลุ่มคำสั่งเมื่อเงื่อนไขที่ตรวจสอบมีค่าเป็นจริง แต่ถ้าเงื่อนไขที่ตรวจสอบเป็นเท็จจะออกจากการทำงานซ้ำ

## รูปแบบคำสั่ง

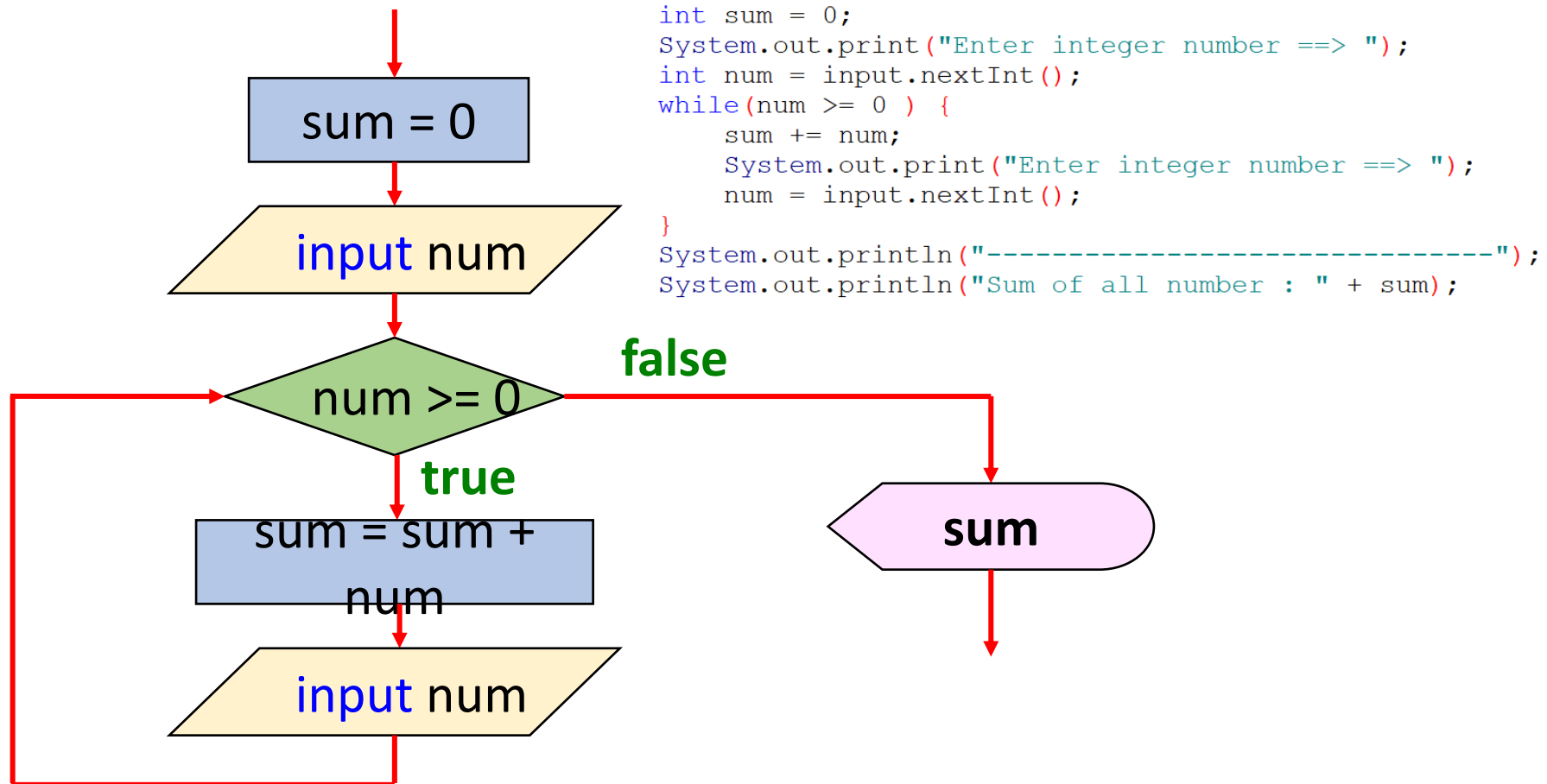
```
while (condition) {  
    statement(s)  
}
```





# คำสั่ง **while** (while statement)

Ex. โปรแกรมบวกเลขจำนวนเต็มสะสม ถ้าป้อนเลขต่ำก

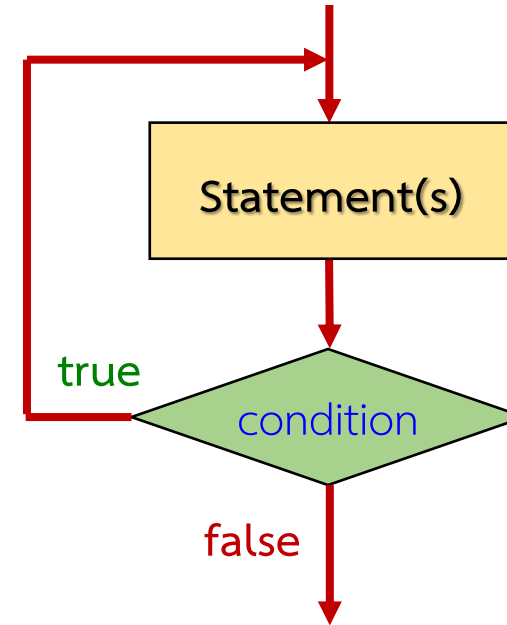


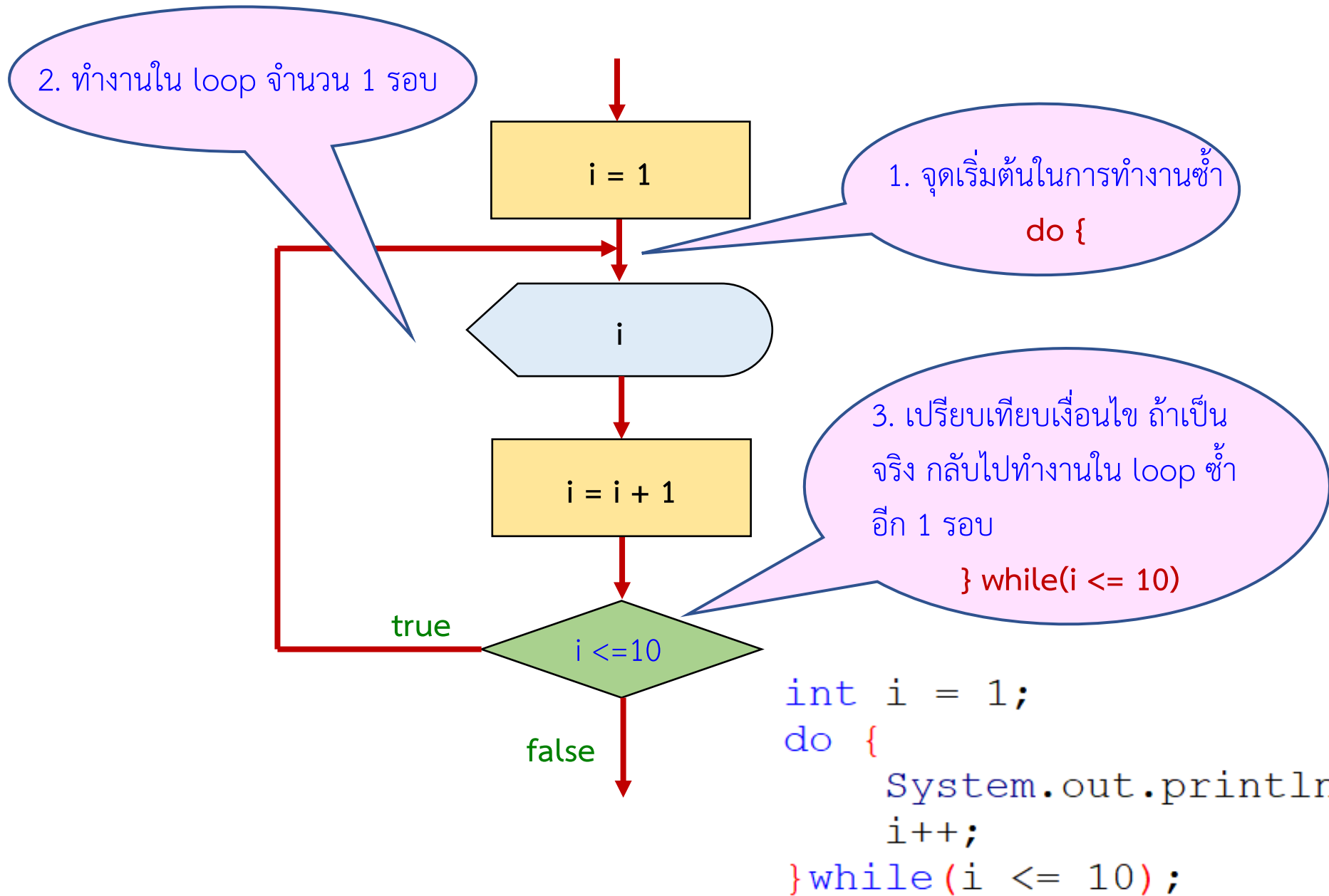
# คำสั่ง **do...while** (do...while statement)

- คำสั่ง **do...while** ใช้สำหรับวนซ้ำทำงานของกลุ่มคำสั่ง โดยให้ทำงานซ้ำ(loop) ก่อนแล้วจึงเปรียบเทียบเงื่อนไข หากเงื่อนไขที่ตรวจสอบมีค่าเป็นจริง ให้ไปทำงานซ้ำอีก 1 รอบ แต่ถ้าเงื่อนไขที่ตรวจสอบเป็นเท็จจะหยุดการทำงานซ้ำ

## รูปแบบคำสั่ง

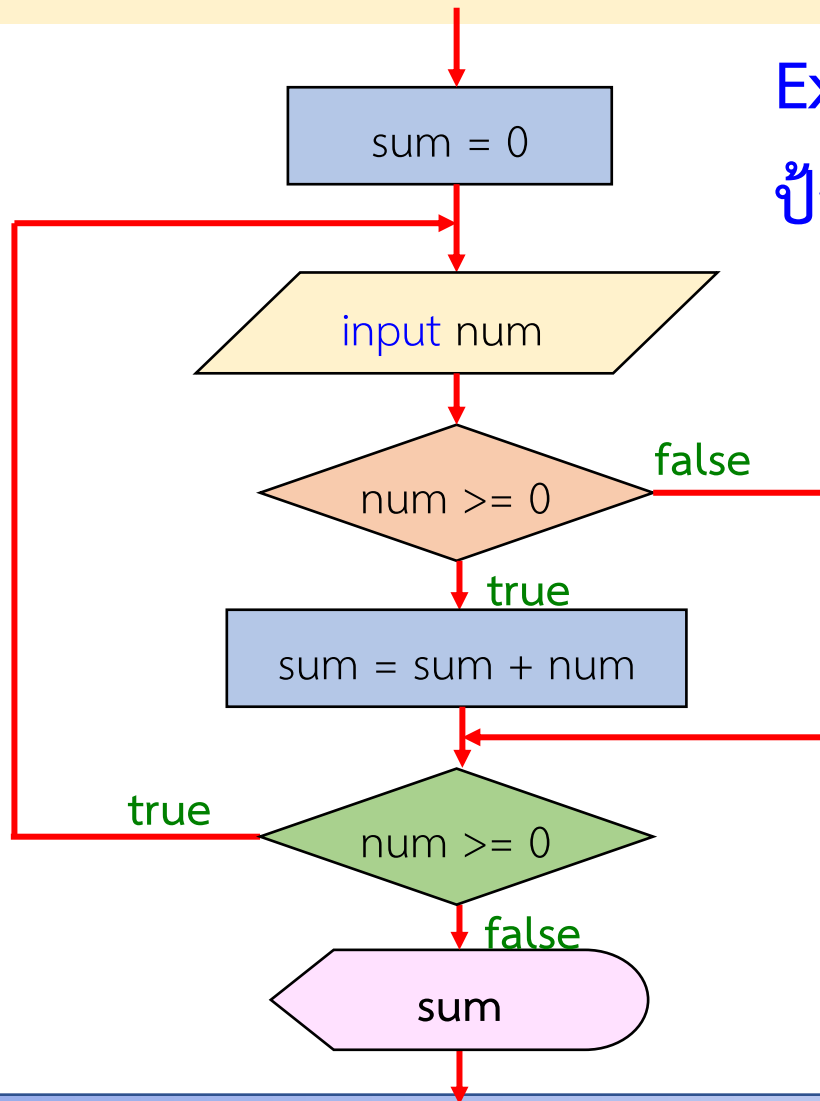
```
do {  
    statement(s)  
} while (condition) ;
```







# คำสั่ง **do...while** (do...while statement)



Ex. โปรแกรมบวกเลขจำนวนเต็มสะสม ถ้า  
ป้อนเลขต่ำกว่า 0 ให้หยุดรับตัวเลข

```
int sum = 0 , num;
do {
    System.out.print("Enter integer number ==> ");
    num = input.nextInt();
    if(num >= 0)
        sum += num;
}while(num >= 0 );

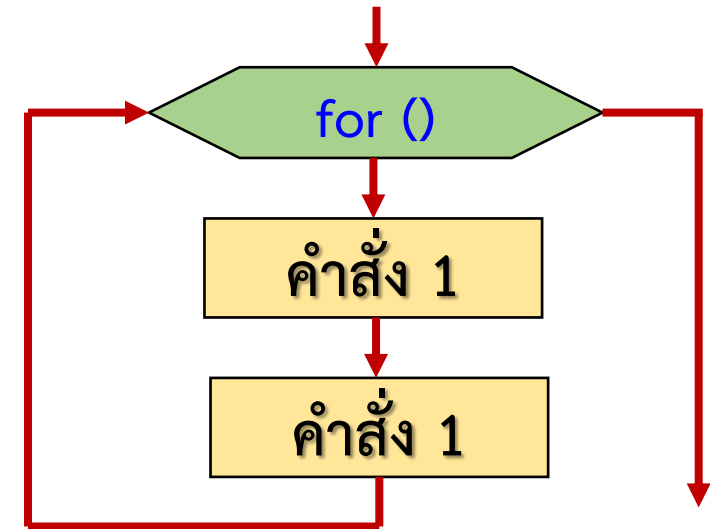
System.out.println("-----");
System.out.println("Sum of all number : " + sum);
```

# คำสั่ง **for** (for statement)

- คำสั่ง **for** ใช้ในการวนซ้ำการทำงานของกลุ่มคำสั่ง โดยจำนวนรอบในการวนซ้ำขึ้นอยู่กับ ค่าเริ่มต้นที่กำหนด ค่าที่เพิ่มขึ้น และค่าสุดท้ายที่ทำให้เงื่อนไขเป็นจริง

## รูปแบบคำสั่ง

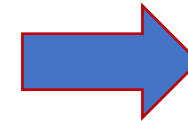
```
for(กำหนดค่าเริ่มต้นให้ตัวแปรนับรอบ; เงื่อนไขที่ตรวจสอบ; ปรับค่าตัวแปรนับรอบ) {  
    คำสั่งที่ให้ทำซ้ำแต่ละรอบ ;  
}
```



# คำสั่ง **for** (for statement)

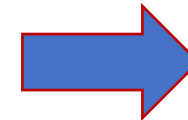
## ตัวอย่าง

```
for (i=1; i<=100; i++)  
{  
    System.out.println(i+1);  
}
```



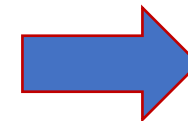
```
2  
3  
4  
...  
101
```

```
for (j=9; j>=0; j--)  
{  
    System.out.println(j);  
}
```



```
9  
8  
7  
...  
0
```

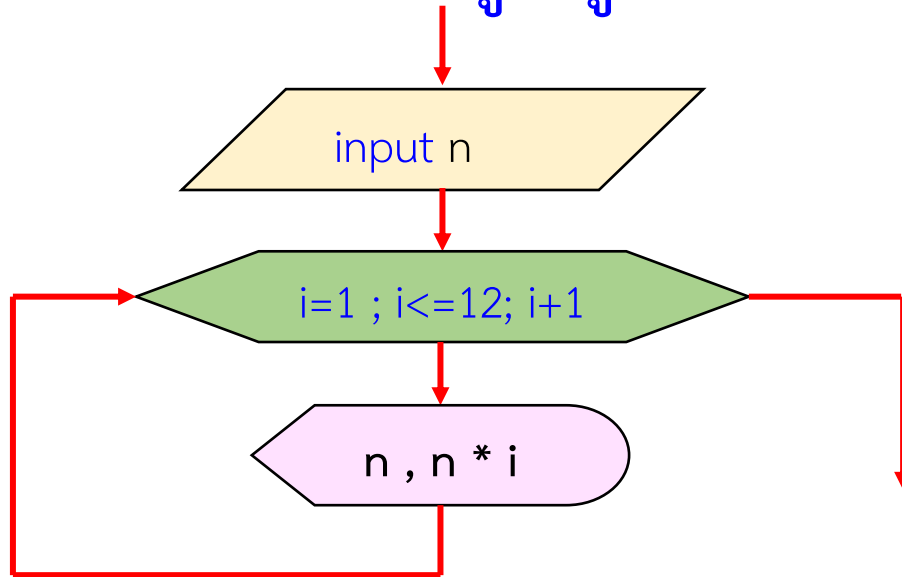
```
for (x=0; x<100; x+=5)  
{  
    System.out.println(x);  
}
```



```
0  
5  
10  
...  
95
```

# คำสั่ง **for** (for statement)

Ex. โปรแกรมแสดงสูตรคูณ 12 บรรทัด ตามแม่สูตรคูณที่กำหนดโดยผู้ใช้



```
Scanner in = new Scanner(System.in);

System.out.print("Enter multiply number ==> ");
int n = in.nextInt();

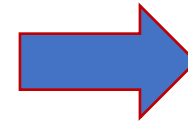
System.out.println("-----");

for(int i=1; i<=12; i++) {
    System.out.format("%2d x %2d = %3d\n",n,i,n*i);
}
```

# คำสั่ง break

- คำสั่ง **break** เป็นคำสั่งให้หยุดและหลุดจาก for, while, do-while หรือ switch โดยเมื่อพบคำสั่งนี้โปรแกรมจะหลุดออกจาก loop เหล่านี้ทันที

```
for (i=1; i<=10; i++)  
{  
    if (i==5)  
        break;  
    System.out.println(i);  
}
```

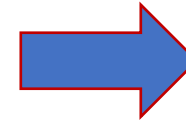


1  
2  
3  
4

# คำสั่ง `continue`

- คำสั่ง `continue` เป็นคำสั่งให้ข้ามการทำงานในรอบการทำงานที่กำหนด แล้วให้ไปทำงานในรอบต่อไป

```
for (i=1; i<=10; i++)  
{  
    if (i==5)  
        continue;  
    System.out.println(i) ;  
}
```



1  
2  
3  
4  
6  
7  
8  
9  
10

# การเขียนลูปซ้อนลูป (Nested loop)

- การเขียนคำสั่งเกี่ยวกับการวนซ้ำซ้อนกัน เช่น

```
for (int i=1; i<=10; i++)  
{  
    for (int j=1; j<=2; j++)  
    {  
        .....  
    }  
}
```

- สามารถเขียนเป็น while ซ้อน while หรือ do ซ้อน do ได้ หรือแบบอื่นๆ ได้

# การเขียนลูปซ้อนลูป (Nested loop)

Ex. โปรแกรมแสดงตารางการคูณ 12 \* 12 ดังรูป

*	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144



```
final int MAX = 12;
//display table header
System.out.print(" * |");
for(int i = 1; i <= MAX; i++) {
    System.out.format("%5d" , i);
}
System.out.println();

for(int i = 1; i <= MAX+1; i++) {
    System.out.print("-----");
}
System.out.println();
```

```
//display table
for(int i = 0; i <= MAX; i++) {
    System.out.format("%3d|" , i);
    for(int j = 1; j <= MAX; j++) {
        System.out.format("%5d" , i*j);
    }
    System.out.println();
}
```

Nested  
Loop

# Arrays

- จัดเก็บข้อมูลหลายค่าที่มีชนิดเดียวกันในตัวแปรเดียว
- ช่วยลดจำนวนตัวแปรที่ต้องประกาศ
- เหมาะกับข้อมูลที่มีลำดับ เช่น คะแนนนักเรียน, จำนวนสินค้า

# Arrays

- รูปแบบ

*// รูปแบบทั่วไป*

```
datatype[] arrayName;  
arrayName = new datatype[size];
```

- หรือ การประกาศอาร์เรย์และการกำหนดค่าเริ่มต้น

```
datatype[] arrayName = new datatype[size];
```

# Arrays

- ตัวอย่าง

```
// Declaration
```

```
int[] numbers;
```

```
String[] names;
```

```
// Initialization
```

```
numbers = new int[5];           // Array of 5 integers
```

```
names = new String[3];         // Array of 3 strings
```

```
// Declaration + Initialization
```

```
int[] scores = new int[10];
```

```
double[] prices = {19.99, 25.50, 12.75};
```

```
String[] cities = {"Bangkok", "Chiang Mai", "Phuket"};
```

# Arrays

- การกำหนดค่าให้อาร์เรย์

```
scores[0] = 80;  
scores[1] = 90;  
scores[2] = 75;
```

- การเข้าถึงข้อมูลในอาร์เรย์

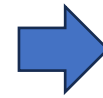
```
System.out.println(scores[1]); // แสดงค่า 90
```

# Arrays

- ตัวอย่างการใช้ for-loop กับ array

```
int[] score = {67, 70, 78, 85, 65, 57, 48};
```

```
for(int i=0; i< score.length; i++) {  
    System.out.print(score[i] + " | ");  
}
```



67 | 70 | 78 | 85 | 65 | 57 | 48 |

- การใช้ Enhanced for-loop

```
int[] score = {67, 70, 78, 85, 65, 57, 48};
```

```
for(int sc : score) {  
    System.out.print(sc + " | ");  
}
```



67 | 70 | 78 | 85 | 65 | 57 | 48 |

# การรับข้อมูลด้วย Scanner

- คลาส Scanner ใช้ในรับข้อมูลจาก console
- ต้องนำเข้าแพ็คเกจ java.util.Scanner

```
import java.util.Scanner ;
```

- สร้าง object ของคลาส Scanner เพื่อรับข้อมูล

ชนิดข้อมูลแบบคลาส  
ชื่อ Scanner

Keyword new ใช้สร้าง object

พารามิเตอร์

```
Scanner in = new Scanner(System.in) ;
```

ตัวแปรแบบ object ชื่อ in

Constructor ของคลาส Scanner

# การรับข้อมูลด้วย Scanner

- การรับข้อมูลด้วยคลาส Scanner จะเป็น String ดังนั้นหากต้องการนำข้อมูลไปใส่ตัวแปรชนิดอื่น ๆ หรือ นำไปคำนวณ ต้องแปลงให้เป็นชนิดข้อมูลที่ต้องการก่อนนำไปใช้ก่อน
- เมธอดในการแปลงข้อมูลที่รับเข้ามาเป็นตัวเลข มีดังนี้

Method	Example
nextByte()	byte b = in.nextByte();
nextDouble()	double d = in.nextDouble();
nextFloat()	float f = in.nextFloat();
nextInt()	int n = in.nextInt();
nextLong()	long l = in.nextLong();
nextShort()	short s = in.nextShort();
next()	String str = in.next();
nextLine()	String str = in.nextLine(); <i>//รับข้อความทั้งบรรทัดรวมช่องว่าง</i>



# การแปลงชนิดข้อมูลจาก String ให้เป็นตัวเลข

ชนิดข้อมูลที่ต้องการ	ตัวอย่าง String	Example
int	<code>String str = "100";</code>	<code>int num = Integer.parseInt(str);</code>
double	<code>String str = "75.50";</code>	<code>double num = Double.parseDouble(str);</code>
float	<code>String str = "75.50";</code>	<code>float num = Float.parseFloat(str);</code>
long	<code>String str = "1234567890";</code>	<code>long num = Long.parseLong(str);</code>

- \*\*\* ถ้า String ไม่ใช่ค่าที่สามารถแปลงเป็นตัวเลขได้ จะเกิดข้อผิดพลาด(Exception)
- หากต้องการแปลงจากตัวเลขเป็น String ก็สามารถใช้ `String.valueOf()` หรือ `+`

# การรับข้อมูลด้วย Scanner

## ตัวอย่าง

## รับค่าจากแป้นพิมพ์ใส่ตัวแปรชนิด String

```
import java.util.Scanner;

public class InputDataScanner {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your name : ");
        String name = input.nextLine();

        System.out.print("Enter you age : ");
        int age = input.nextInt();

        System.out.println("Hello, "+name+"! You are "+age+" years old.");
        input.close();
    }
}
```

Enter your name : *Nidnoi*  
Enter you age : *18*  
Hello, Nidnoi! You are 18 years old.

# การรับข้อมูลด้วย Scanner

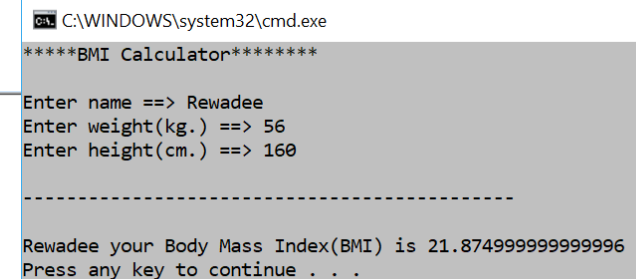
## ตัวอย่าง

## รับค่าจากแป้นพิมพ์ใส่ตัวแปรชนิดตัวเลข

```
import java.util.Scanner;

public class BmiCalculator {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        String name;
        double w , hc ,hm , bmi ;
        System.out.println("*****BMI Calculator*****\n");
        System.out.print("Enter name ==> ");
        name = in.next();
        System.out.print("Enter weight(kg.) ==> ");
        w = in.nextDouble();
        System.out.print("Enter height(cm.) ==> ");
        hc = in.nextDouble();

        hm = hc /100;
        bmi = w / Math.pow(hc,2);
        System.out.println("\n-----\n");
        System.out.println(name + " your Body Mass Index(BMI) is " + bmi);
    }
}
```



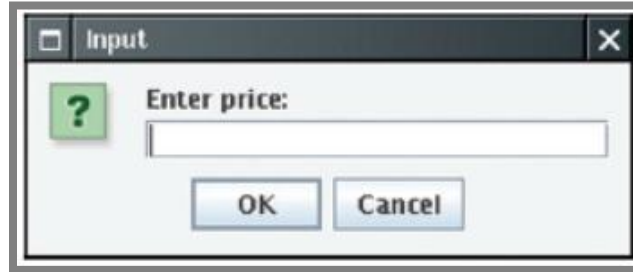
```
C:\WINDOWS\system32\cmd.exe
*****BMI Calculator*****
Enter name ==> Rewadee
Enter weight(kg.) ==> 56
Enter height(cm.) ==> 160

-----

Rewadee your Body Mass Index(BMI) is 21.874999999999996
Press any key to continue . . .
```

## \*\* การนำเข้าข้อมูลผ่าน Dialog Box (JOptionPane) \*\*

- คลาส JOptionPane ใช้ในการรับข้อมูลเข้าผ่าน Dialog Box



- ให้เขียน `import javax.swing.JOptionPane;` ไว้ที่บรรทัดแรก
- เรียกใช้เมธอด `showInputDialog`

```
String input = JOptionPane.showInputDialog(prompt);
```

- โดยทั่วไปการรับข้อมูลเข้าผ่านทาง JOptionPane จะรับเข้ามาเป็น String ดังนั้นหากต้องการให้ข้อมูลเปลี่ยนเป็นตัวเลข จะต้อง Convert ข้อมูล

```
int count = Integer.parseInt(input);  
Double price = Double.parseDouble(input);
```

## **\*\* การนำเข้าข้อมูลผ่าน Dialog Box (JOptionPane) \*\***

- เมื่ออ่านข้อมูลมาแล้วต้องการแสดงข้อมูลที่อ่านเข้ามาผ่านทาง JOptionPane เช่นเดิม จะเรียกใช้เมธอด `showMessageDialog`

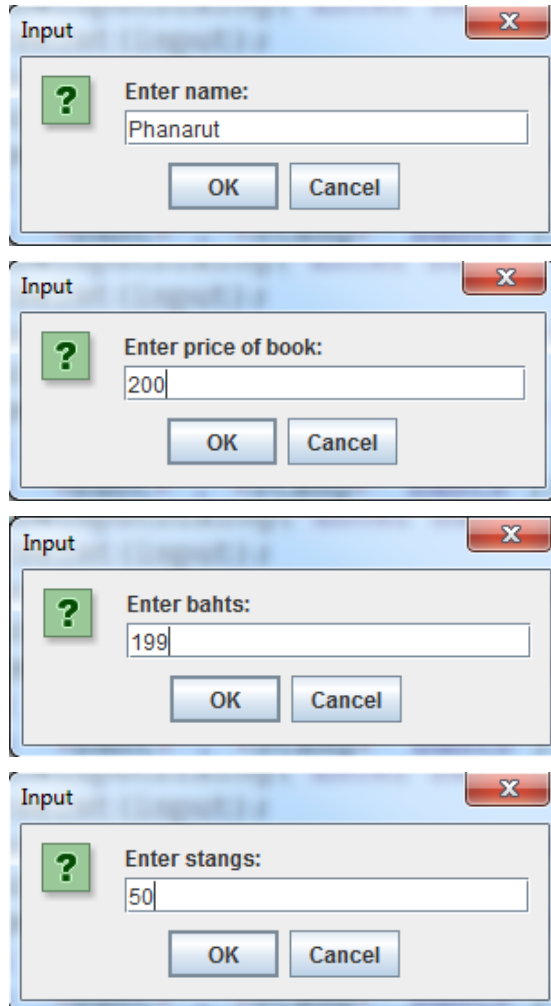
```
JOptionPane.showMessageDialog(null, "Your change is 3.05");
```



## **\*\* การนำเข้าข้อมูลผ่าน Dialog Box (JOptionPane) \*\***

```
import javax.swing.JOptionPane;
public class Test
{ public static void main(String[] args)
{
    String name;
    name = JOptionPane.showInputDialog("Enter name: ");
    String input;
    input = JOptionPane.showInputDialog("Enter price of book: ");
    double price = Double.parseDouble(input);
    input = JOptionPane.showInputDialog("Enter bahts: ");
    int baht = Integer.parseInt(input);
    input = JOptionPane.showInputDialog("Enter stangs: ");
    int stang = Integer.parseInt(input);
    JOptionPane.showMessageDialog(null, "Your name is " + name+"\n"+
        "The price of book is "+price+" bahts\n"+
        "Your money is "+baht+"."+stang+" bahts");
}
}
```

## \*\* การนำเข้าข้อมูลผ่าน Dialog Box (JOptionPane) \*\*



The image shows four sequential 'Input' dialog boxes, each with a green question mark icon and 'OK'/'Cancel' buttons. The first box prompts 'Enter name:' with the text 'Phanarut'. The second box prompts 'Enter price of book:' with the value '200'. The third box prompts 'Enter bahts:' with the value '199'. The fourth box prompts 'Enter stangs:' with the value '50'.

Input

Enter name:  
Phanarut

OK Cancel

Input

Enter price of book:  
200

OK Cancel

Input

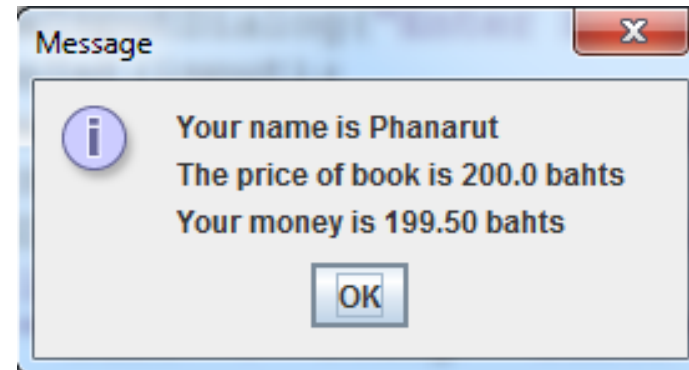
Enter bahts:  
199

OK Cancel

Input

Enter stangs:  
50

OK Cancel



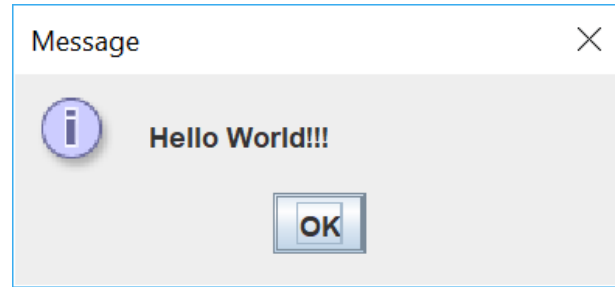
The image shows a 'Message' dialog box with an information icon (i) and an 'OK' button. It displays the collected data: 'Your name is Phanarut', 'The price of book is 200.0 bahts', and 'Your money is 199.50 bahts'.

Message

Your name is Phanarut  
The price of book is 200.0 bahts  
Your money is 199.50 bahts

OK

# การแสดงผลผ่าน Dialog box (JOptionPane)



ต้องนำเข้าแพ็คเกจและคลาสในการใช้งานดังนี้

```
import javax.swing.JOptionPane;
```

เรียกใช้เมธอด `showMessageDialog()` ในการแสดง dialog

```
JOptionPane.showMessageDialog(null, "Hello World!!!");
```

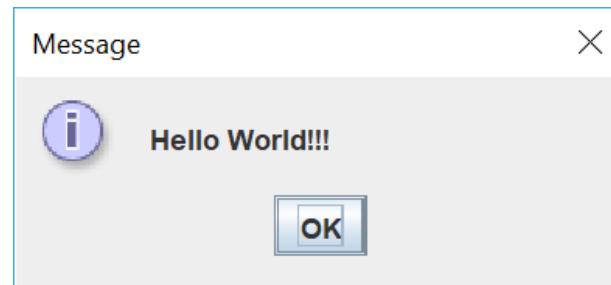


# การแสดงผลผ่าน Dialog box (JOptionPane)

## ตัวอย่าง

```
import javax.swing.JOptionPane;

public class DisplayDialog {
    public static void main(String args[]) {
        JOptionPane.showMessageDialog(null, "Hello World!!!");
    }
}
```



# การจัดรูปแบบ Output

- ตัวอย่างโค้ด Java

```
// Using printf  
int age = 25;  
double salary = 50000.75;  
String name = "John";  
  
System.out.printf("Name: %s\n", name);  
System.out.printf("Age: %d years old\n", age);  
System.out.printf("Salary: $%.2f\n", salary);  
  
// Format specifiers  
// %s - String  
// %d - Integer  
// %f - Float/Double  
// %.2f - 2 decimal places  
// %n - New line (platform independent)
```

# การจัดรูปแบบ Output

- ตัวอย่างโค้ด Java

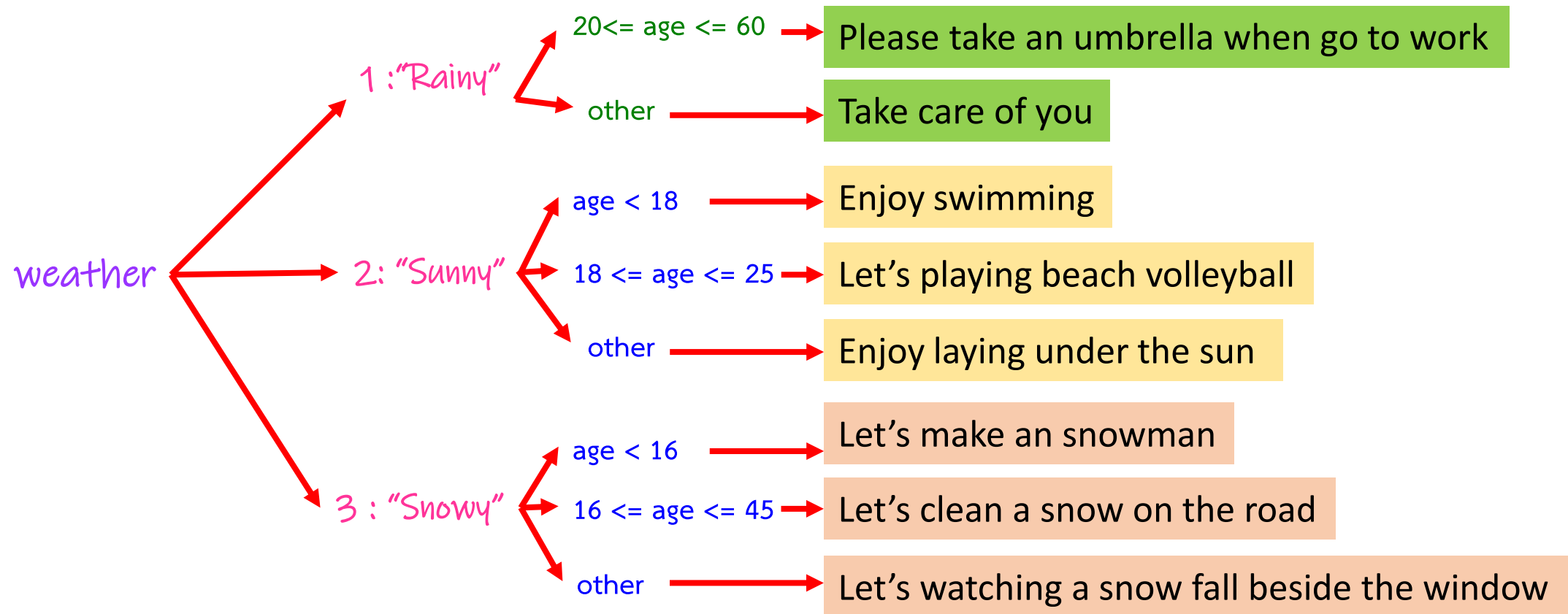
```
String message = String.format("Hello %s, you are %d years old", name, age);  
System.out.println(message);
```

```
// Padding and alignment
```

```
System.out.printf("%-10s %5d %8.2f\n", "John", 25, 1234.56);  
System.out.printf("%-10s %5d %8.2f\n", "Alice", 30, 2500.00);
```

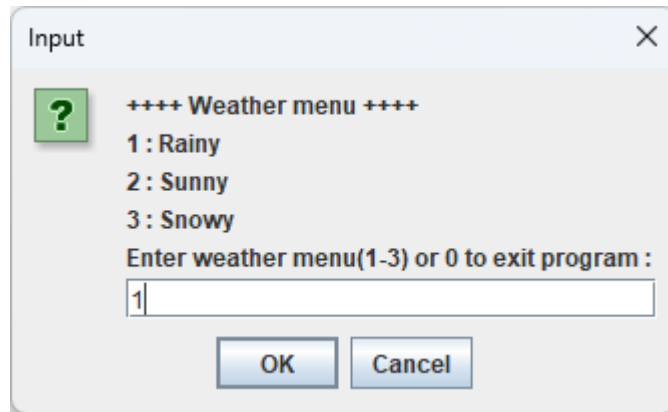
แบบฝึกหัด

Ex1. ให้เขียนโปรแกรมรับค่าสภาพอากาศ(Rainy/Sunny/Snowy) และอายุของผู้ใช้ เพื่อแนะนำกิจกรรมที่เหมาะสม โดยมีเงื่อนไขตามต้นไม้ตัดสินใจ(decision tree) ดังนี้

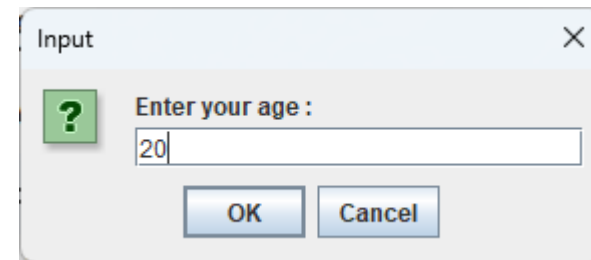


Ex1. ให้เขียนโปรแกรมรับค่าสภาพอากาศ(Rainy/Sunny/Snowy) และอายุของผู้ใช้ เพื่อแนะนำกิจกรรมที่เหมาะสม โดยมีเงื่อนไขตามต้นไม้ตัดสินใจ(decision tree) ดังนี้

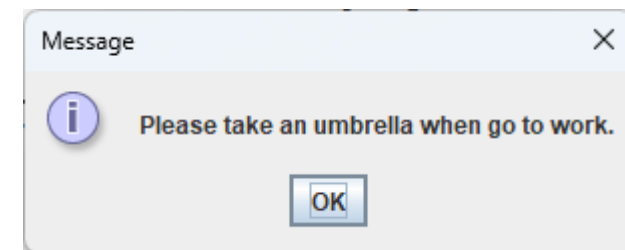
ตัวอย่างผลลัพธ์



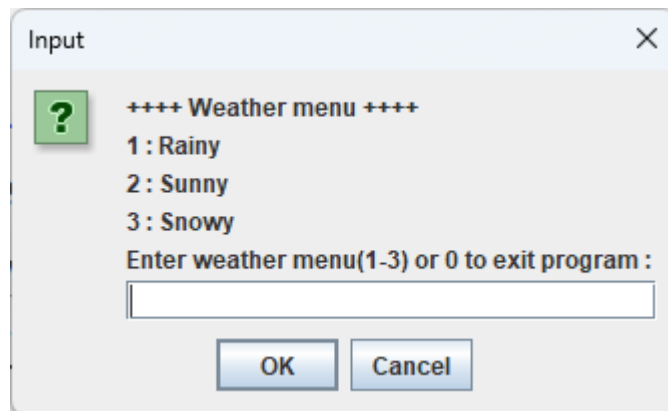
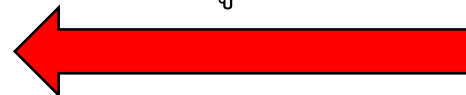
ถ้าป้อนเมนู 1-3 จะรับอายุ



แสดงผลกิจกรรมตามเงื่อนไข

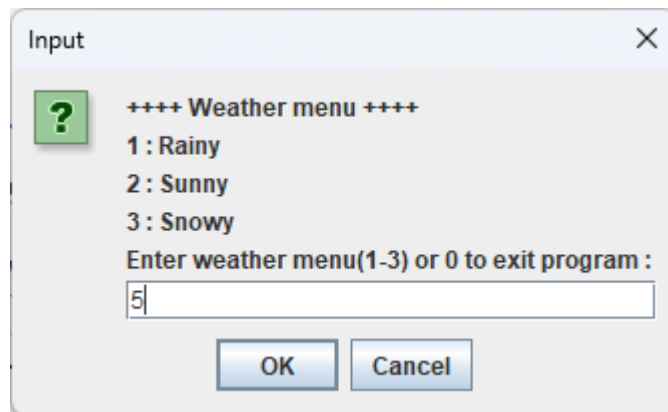


วนไปแสดงเมนูและรับค่าอีกครั้ง

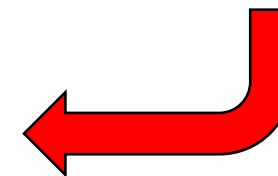
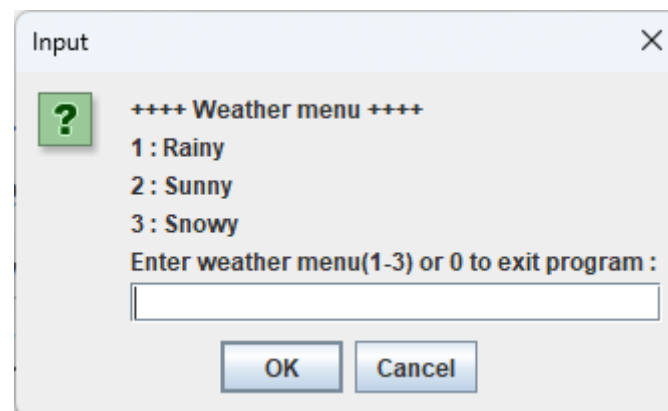
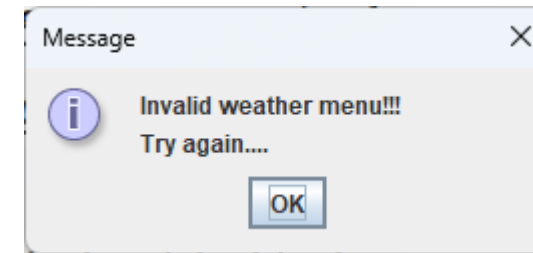


Ex1. ให้เขียนโปรแกรมรับค่าสภาพอากาศ(Rainy/Sunny/Snowy) และอายุของผู้ใช้ เพื่อแนะนำกิจกรรมที่เหมาะสม โดยมีเงื่อนไขตามต้นไม้ตัดสินใจ(decision tree) ดังนี้

ตัวอย่างผลลัพธ์



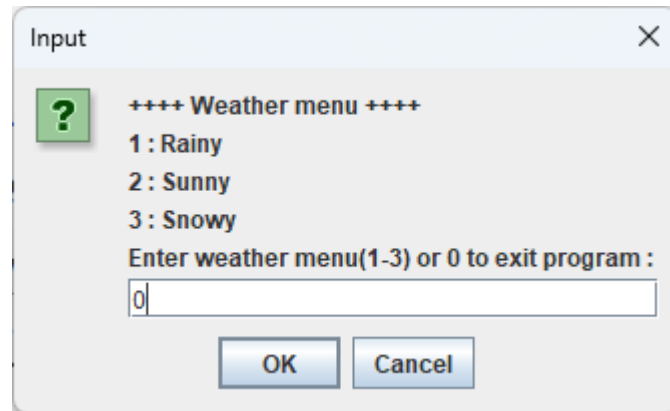
ถ้าป้อนเมนูนอกเหนือจาก  
0 - 3



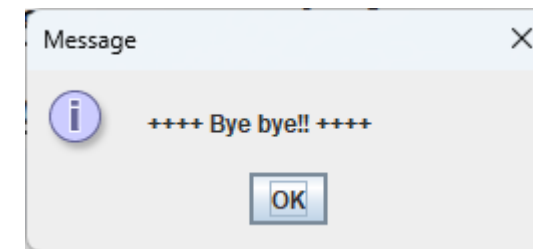
วนไปแสดงเมนูและรับค่าอีกครั้ง

Ex1. ให้เขียนโปรแกรมรับค่าสภาพอากาศ(Rainy/Sunny/Snowy) และอายุของผู้ใช้ เพื่อแนะนำกิจกรรมที่เหมาะสม โดยมีเงื่อนไขตามต้นไม้ตัดสินใจ(decision tree) ดังนี้

ตัวอย่างผลลัพธ์



ถ้าป้อนเมนู 0



จบโปรแกรม



Ex1. ให้เขียนโปรแกรมรับค่าสภาพอากาศ(Rainy/Sunny/Snowy) และอายุของผู้ใช้ เพื่อแนะนำกิจกรรมที่เหมาะสม โดยมีเงื่อนไขตามต้นไม้ตัดสินใจ(decision tree) ดังนี้

ข้อกำหนดให้แยกการทำงานเป็นเมธอด(ฟังก์ชัน) ประกอบด้วย

- เมธอด main()
- เมธอดแสดงเมนูการทำงาน
- เมธอดสำหรับแสดงผลกิจกรรมตามเงื่อนไข

## Ex2. โปรแกรมคำนวณเกรด โดยการรับชื่อนักศึกษาเก็บไว้ในอาร์เรย์ รับคะแนนสอบกลางภาคและปลายภาคเพื่อคำนวณคะแนนรวมเก็บไว้ในอาร์เรย์ แล้วให้แสดงข้อมูลชื่อ, คะแนนสอบรวมและเกรดที่ได้

```
++++ Grade calculation program ++++
Enter number of student => 3
+++++
Student no.1
    Student name ==> Rewadee Piputsoongnern
    Midterm score ==> 54
    Final score ==> 27
-----
Student no.2
    Student name ==> Kunawut Boonkhwang
    Midterm score ==> 45
    Final score ==> 22
-----
Student no.3
    Student name ==> Khwanchai Suksean
    Midterm score ==> 40
    Final score ==> 18
-----
```



+++++++ Student grade report ++++++			
-----			
Student name		Score	Grade
-----			
Rewadee	Piputsoongnern	81.00	A
Kunawut	Boonkhwang	67.00	C+
Khwanchai	Suksean	58.00	D+

ข้อกำหนดให้แยกการทำงานเป็นเมธอด(ฟังก์ชัน) ประกอบด้วย

- เมธอด main()
- เมธอดสำหรับคำนวณคะแนนรวม
- เมธอดสำหรับคำนวณเกรด
- เมธอดสำหรับแสดงรายงานเกรด

# END.

# Q & A