

# Part



4

Java Programming Language  
Mr.Rungrote Phonkam  
rungrote@it.kmitl.ac.th



# Contents

1. Statements
2. Comments
3. Operators



# 1. Statements

## คำสั่งหรือสเตจเมนต์

- ใช้กำหนดตัวแปร, ประมวลผล หรือสร้างเอ็กเพลสชัน, ...
- ปิดท้ายด้วย ;

ตัวอย่าง 1      `int i;`  
                  `System.out.print("Hello");`

ตัวอย่าง 2      `float f; f = 12.5; Integer id = new Integer(123);`

ตัวอย่าง 3      `a = (a < 0)?`  
                  `a * -1 : a;`



## 2. Comments

### หมายเหตุหรือคอมเมนต์

สำหรับใส่ข้อความใดๆ เช่นผู้สร้าง วันที่ คำอธิบาย

### รูปแบบ (สำหรับข้อความสั้น)

```
// Text_Comment
```

### เมื่อ

*Text\_Comment* คือข้อความใดๆที่ต้องการสร้างเพื่อเป็นคอมเมนต์

### ตัวอย่าง 1

```
// การคำนวณผลรวม
```

### ตัวอย่าง 2

```
a = (a<0)? a * -1 : a; // การหาค่าสัมบูรณ์
```



## 2. Comments

หมายเหตุหรือคอมเมนต์

รูปแบบ (สำหรับข้อความยาว)

```
/* Text_Comment  
*/
```

เมื่อ

*Text\_Comment* คือข้อความใดๆที่ต้องการสร้างเพื่อเป็นคอมเมนต์

ตัวอย่าง 1

```
/* คอมเมนต์ใช้สำหรับข้อความหนึ่งบรรทัด */
```

ตัวอย่าง 2

```
/* ผู้เขียนโปรแกรม: นายชยันดีบั๊กโปรแกรม  
วันที่สร้าง: 9/9/99  
วันที่แก้ไขโปรแกรมล่าสุด: 10/9/99 */
```



# 3 Operators

## ตัวดำเนินการ

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)
- ตัวดำเนินการข้อมูลเชิงบิต (Bitwise Operators)
- ตัวดำเนินการข้อมูลตรรกะ (Boolean Operators)
- ตัวดำเนินการสำหรับข้อความ (String Operators)



# 3.1. Arithmetic Operators

- ประเภทการคำนวณ

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
+	การบวก	10+5	15
-	การลบ	10-5	5
*	การคูณ	10*5	50
/	การหาร (ได้ผลหาร)	10/5	2
%	การหาร (ได้เศษจากการหาร)	10%5	0



# 3.1. Arithmetic Operators

```
class TriangleArea
{
    public static void main(String args[])
    {
        float area;
        int base = Integer.parseInt(args[0]);
        int height = Integer.parseInt(args[1]);
        area = 1/2 * base * height;
        System.out.println("Area of Triangle is: " + area);
    }
}
```

```
java TriangleArea 15 22
Area of Triangle is: 0.0
```





# 3.1. Arithmetic Operators

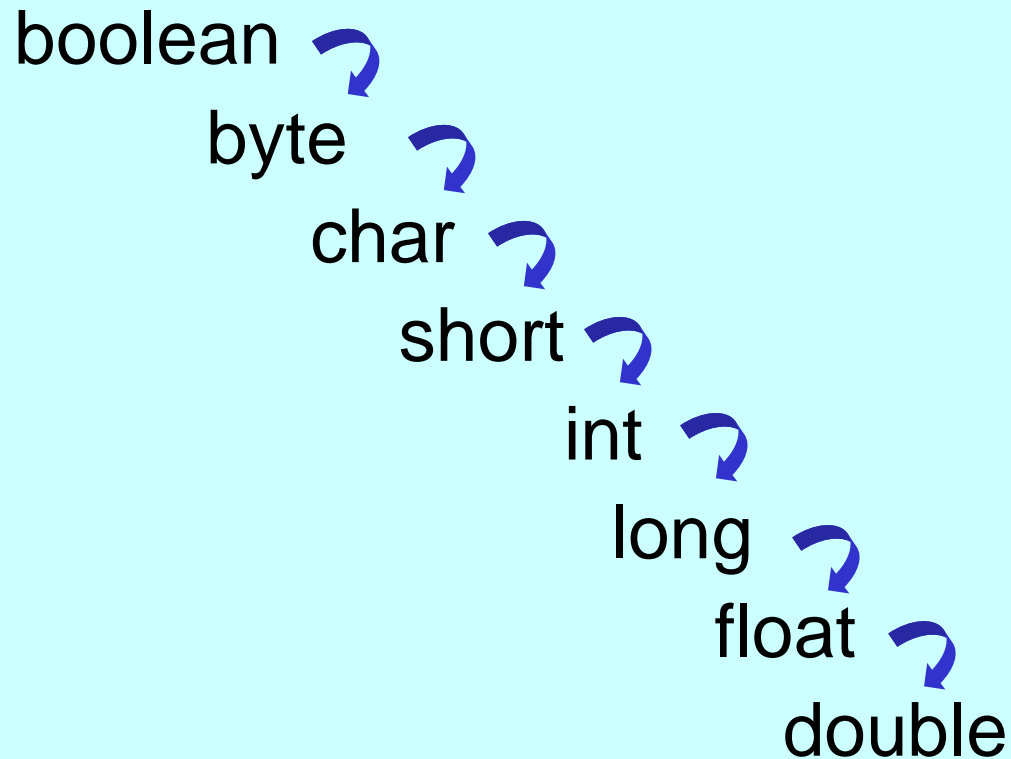
1/2	ได้ผลลัพธ์	0
1.0/2	ได้ผลลัพธ์	0.5
1.0/2.0	ได้ผลลัพธ์เป็น	0.5

Why ???



# 3.1. Arithmetic Operators

## Type Promotion





# 3.1. Arithmetic Operators

- ประเภทการคำนวณและให้ค่า

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	เปรียบได้กับ	ผลลัพธ์ (เมื่อ a = 10)
+=	การบวก	a +=5	a = a + 5	15
-=	การลบ	a -=5	a = a - 5	5
*=	การคูณ	a *=5	a = a * 5	50
/=	การหาร (ได้ผลหาร)	a /=5	a = a / 5	2
%=	การหาร (ได้เศษจากการหาร)	a %=5	a = a % 5	0



# 3.1. Arithmetic Operators

```
class RectangleArea
{
    public static void main(String args[])
    {
        float a = Float.parseFloat(args[0]);
        a *= a;
        System.out.println("Area of Rectangle is : " + a);
    }
}
```

```
java RectangleArea 12
Area of Rectangle is : 144.0
```



# 3.1. Arithmetic Operators

- ประเภทการคำนวณเพิ่มค่าและลดค่าข้อมูล

ตัวดำเนินการ ความหมาย

++Variable	เพิ่มค่าหนึ่งค่าก่อนการเข้าถึง		
		System.out.print(++a)	11 11
Variable++	เพิ่มค่าหนึ่งค่าหลังการเข้าถึง		
		System.out.print(a++)	10 11
--Variable	ลดค่าหนึ่งค่าก่อนการเข้าถึง		
		System.out.print(--a)	9 9
Variable--	ลดค่าหนึ่งค่าหลังการเข้าถึง		
		System.out.print(a--)	10 9



# 3.1. Arithmetic Operators

```
class IncreaseYourNumber
{
    public static void main(String args[])
    {
        int a = Integer.parseInt(args[0]);
        System.out.println("Your Number is " + (a++) + "
            Increase to " + a );
    }
}
```

```
java IncreaseYourNumber 99
Your Number is 99 Increase to 100
```



## 3.2. Bitwise Operators

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์ของ a
			(เมื่อ a = 1101; b = 1001)
~	NOT (1 เป็น 0; 0 เป็น 1)	~a	0010
&	AND	a & b	1001
	OR	a   b	1101
<<	ขยับบิตทางซ้าย	a << 2	0100
>>	ขยับบิตทางขวา	a >> 2	0011
>>>	ขยับบิตทางขวา	(สำหรับข้อมูล unsign) a >>> 2	



## 3.2. Bitwise Operators

```
class ComplementBit
{ public static void main(String args[])
  { int a = Integer.parseInt(args[0]);
    System.out.println("a \t= " + Integer.toBinaryString(a));
    System.out.println("\t-----");
    System.out.println("~a\t= " + Integer.toBinaryString(~a));
    System.out.println("\t=====");
  }
}
```

```
java ComplementBit 124
a      = 1111100
      -----
~a     = 11111111111111111111111111110000011
      =====
```





## 3.2. Bitwise Operators

```
class ANDBit
{ public static void main(String args[])
  { int a = Integer.parseInt(args[0]);
    int b = Integer.parseInt(args[1]);
    System.out.println("a \t= " + Integer.toBinaryString(a));
    System.out.println("b \t= " + Integer.toBinaryString(b));
    System.out.println("\t-----");
    System.out.println("a & b\t= " + Integer.toBinaryString(a & b));
    System.out.println("\t=====");
  }
}
```

```
java ANDBit 100 200
a      = 1100100
b      = 11001000
-----
a & b  = 1000000
```



## 3.2. Bitwise Operators

```
class ORBit
{ public static void main(String args[])
  { int a = Integer.parseInt(args[0]);
    int b = Integer.parseInt(args[1]);
    System.out.println("a \t= " + Integer.toBinaryString(a));
    System.out.println("b \t= " + Integer.toBinaryString(b));
    System.out.println("\t-----");
    System.out.println("a | b\t= " + Integer.toBinaryString(a | b));
    System.out.println("\t=====");
  }
}
```

```
java ORBit 16 40
```

```
a      = 10000
```

```
b      = 101000
```

```
-----
```

```
a | b  = 111000
```



## 3.2. Bitwise Operators

```
class ShiftLeftBit
{   public static void main(String args[])
    {       int a = 1;
            System.out.println("2 Power 1 = " + a);
            System.out.println("2 Power 2 = " + (a<<1));
            System.out.println("2 Power 3 = " + (a<<2));
            System.out.println("2 Power 4 = " + (a<<3));
            System.out.println("2 Power 5 = " + (a<<4));
    }
}
```

**java ShiftLeftBit**

2 Power 1 = 1

2 Power 2 = 2

2 Power 3 = 4

2 Power 4 = 8

2 Power 5 = 16



## 3.2. Bitwise Operators

```
class ShiftRightBit
{   public static void main(String args[])
    {       int a = 10;
            System.out.println("2 Power 1 = " + a);
            System.out.println("2 Power 2 = " + (a>>>1));
            System.out.println("2 Power 3 = " + (a>>>2));
            System.out.println("2 Power 4 = " + (a>>>3));
            System.out.println("2 Power 5 = " + (a>>>4));
    }
}
```

```
java ShiftRightBit
2 Power 1 = 10
2 Power 2 = 5
2 Power 3 = 2
2 Power 4 = 1
2 Power 5 = 0
```



## 3.3. Boolean Operators

- ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ ความหมาย

ตัวอย่าง ผลลัพธ์ของ a

(เมื่อ a = 10; b = 15)

== ค่าเท่ากันหรือไม่

a == b False (เท็จ)

!= ค่าไม่เท่ากันหรือไม่

a != b True (จริง)

> ค่ามากกว่าหรือไม่

a > b False (เท็จ)

>= ค่ามากกว่าหรือเท่ากันหรือไม่

a >= b False (เท็จ)

< ค่าน้อยกว่าหรือไม่

a < b True (จริง)

<= ค่าน้อยกว่าหรือเท่ากันหรือไม่

a <= b True (จริง)



## 3.3. Boolean Operators

```
class TrueOrFalse
{ public static void main(String args[])
  {
    System.out.println(true);
    System.out.println(true==false);
    System.out.println(3.00==3);
    System.out.println('a'!='A');
    int a = 120, b = 300;
    System.out.println(a > b);
    System.out.println(a/b < 0);
  }
}
```

```
java TrueOrFalse
true
false
true
true
false
false
```



## 3.3. Boolean Operators

- ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์ของ a
		(เมื่อ a = 10; b = 15)	
!	กลับค่าทางตรรกะ	!(a == b)	true (จริง)
&&	AND ค่าทางตรรกะ	(5 >= a) && (5 <= b)	false (เท็จ)
	OR ค่าทางตรรกะ	(15 == a)    (15 == b)	true (จริง)



## 3.3. Boolean Operators

```
class ANDBoolean
{
    public static void main(String args[])
    {
        boolean bl1 = false, bl2 = false;
        System.out.println(bl1 + " && " + bl2 + " = " + (bl1 && bl2));
        System.out.println(bl1 + " && " + !bl2 + " = " + (bl1 && !bl2));
        System.out.println(!bl1 + " && " + bl2 + " = " + (!bl1 && bl2));
        System.out.println(!bl1 + " && " + !bl2 + " = " + (!bl1 && !bl2));
    }
}
```

```
java ANDBoolean
false && false = false
false && true = false
true && false = false
true && true = true
```





## 3.3. Boolean Operators

```
class ORBoolean
{ public static void main(String args[])
  { boolean bl1 = false, bl2 = false;
    System.out.println(bl1 + " || " + bl2 + " = " + (bl1 || bl2));
    System.out.println(bl1 + " || " + !bl2 + " = " + (bl1 || !bl2));
    System.out.println(!bl1 + " || " + bl2 + " = " + (!bl1 || bl2));
    System.out.println(!bl1 + " || " + !bl2 + " = " + (!bl1 || !bl2));
  }
}
```

```
java ORBoolean
false || false = false
false || true = true
true || false = true
true || true = true
```



## 3.4. String Operators

### การดำเนินการสำหรับข้อความ

- การเชื่อมข้อความ ใช้เครื่องหมาย “+” ในการเชื่อมข้อความได้  
เช่น “Hello “ + “Java” มีค่าเท่ากับ “Hello Java”
- การดำเนินการกับข้อความใดๆใน Java สามารถใช้เมธอดของ  
คลาส String เช่น

parseInt	แปลงข้อความตัวเลขให้เป็นค่าตัวเลข
charAt	ดึงตัวอักษรในข้อความณ.ตำแหน่งใดๆ
length	หาความยาวของตัวอักษรในข้อความ
substring	ตัดบางส่วนของข้อความมาใช้งาน
toLowerCase	แปลงตัวอักษรในข้อความให้เป็นตัวเล็ก
...	...



## 3.4. String Operators

```
class ShowString
{ public static void main(String args[])
  {
    System.out.println(args[0] + args[1] + args[2]);
  }
}
```

```
java ShowStrng Hi Java World
Hi Java World
```