

บทที่ 2

ข้อมูลชนิดพื้นฐานและตัวดำเนินการ

2.1 ข้อมูลชนิดพื้นฐาน (Primitive data type)

Primitive data types เป็นชนิดของตัวแปรพื้นฐานที่ไม่สามารถแตกย่อยได้อีก Java programming สนับสนุน 8 primitive data types ดังนี้

ตารางที่ 2.1 ช่วงการเก็บข้อมูลของ Primitive data type แต่ละชนิด

ชนิด	ขนาด(bits)	Default	Min_value	Max_value
byte	8	0	-128	127
short	16	0	-32,768	32,767
int	32	0	-2,147,483,648	2,147,483,647
long	64	0	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
float	32	0.0	-3.40292347E+38	3.4028237E+38
double	64	0.0	-1.79769313486231570E+308	1.79769313486231570E+308
boolean	1	False	-	-
char	16	'\u0000'	'\u0000' or 0	'\uffff' or 65,535

จากตารางที่ 2.1 เราสามารถแบ่ง Primitive types เป็น 4 กลุ่ม ได้ดังนี้

1) **Integer** คือ byte, short, int และ long ตัวแปรเหล่านี้เป็นตัวแปรของชนิดข้อมูลจำนวนเต็ม โดยสามารถเก็บค่าที่เป็นบวกหรือลบได้

2) **Real Number** คือ float และ double เป็นตัวแปรของมูลจำนวนจริงที่สามารถอยู่ในรูปทศนิยม โดยส่วนใหญ่จะเป็นข้อมูลทางวิทยาศาสตร์หรือทางคณิตศาสตร์ที่ต้องการความละเอียดของข้อมูลมาก ๆ เช่น float PI = 3.14159265359

3) **Truth Value** คือ boolean มีค่าเป็น true หรือ false เท่านั้น

4) **Character** คือ ชนิดข้อมูล char สำหรับเก็บตัวอักษรที่อยู่ในรูป Unicode ซึ่ง Unicode เป็นรูปแบบตัวอักษรสากล และเก็บตัวอักษรสำหรับภาษาของส่วนใหญ่ที่ใช้ในปัจจุบัน

รูปแบบการประกาศตัวแปร Primitive data type

dataType VarName [=Value];	
dataType	ชนิดของตัวแปร เช่น int, float, double เป็นต้น
VarName	ชื่อตัวแปร
[=Value]	เป็นการกำหนดค่าให้กับตัวแปร สัญลักษณ์ [] เป็น option มีหรือไม่ก็ได้
ตัวอย่าง	int a; ประกาศ a เป็นตัวแปรชนิดจำนวนเต็ม float f=10.25; ประกาศ f เป็นตัวแปรทศนิยมมีค่าเท่ากับ 10.25

<pre> 1 class TestPrimitive 2 { 3 public static void main(String[] args) 4 { 5 Boolean result = true; 6 char capitalC = 'C'; 7 byte b = 100; 8 short s = 10000; 9 int i = 100000; 10 int decVal = 26; 11 int octVal = 032; 12 int hexVal = 0x1a; 13 double d1 = 123.4, d2 = 1.234e2; 14 float f1 = 123.4f; 15 System.out.println("result: "+ result); 16 System.out.println("capital: "+ capitalC); 17 System.out.println("b: "+ b); 18 System.out.println("s: "+ s); 19 System.out.println("i: "+ i); 20 System.out.println("octVal: "+ octVal); 21 System.out.println("hexVal: "+ hexVal); 22 System.out.println("d1: "+ d1); 23 System.out.println("d2: "+ d2); 24 System.out.println("f1: "+ f1); 25 } 26 }</pre>	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5-14 เป็นการประกาศตัวแปรชนิดพื้นฐานแบบต่างๆ พร้อมกำหนดค่า</p> <p>บรรทัดที่ 15-24 แสดงผลลัพธ์ของตัวแปรด้วยคำสั่ง System.out.println()</p>
<pre> 1 ---- exec: TestPrimitive 2 3 result: true 4 capital: C 5 b: 100 6 s: 10000 7 i: 100000 8 octVal: 26 9 hexVal: 26 10 d1: 123.4 11 d2: 123.4 12 f1: 123.4 13 14 ---- operation: complete</pre>	

2.2 ตัวแปรชุด (Array)

Array คือ ตัวแปรชุดของตัวแปรชนิดเดียวกัน โดยสามารถใช้ได้ทั้งตัวแปร primitive และตัวแปรแบบ reference data การอ้างค่าในตัวแปร array จะอ้างผ่าน index ซึ่ง index จะเริ่มตั้งแต่ 0 ตัวอย่างเช่น array ของตัวแปรจำนวนเต็ม x ขนาด 10 ตัว แสดงดังรูปที่ 2.1 โดยมี index คือ 0, 1, 2,...,9

	X[0]	X[1]	X[2]	X[3]	X[4]	X[5]	X[6]	X[7]	X[8]	X[9]
X[10]	8	12	5	2	6	42	7	5	81	22
Index	0	1	2	3	4	5	6	7	8	9

รูปที่ 2.1 รูปแสดงตัวแปรชุด $x[10]$ โดย index มีค่าตั้งแต่ 0-9

รูปแบบการประกาศตัวแปร Array

แบบที่ 1		แบบที่ 2	
	<code>dataType[] ArrayName;</code> <code>ArrayName = new dataType[n]</code>		<code>dataType[] ArrayName = new dataType[n];</code>
dataType[]	ชนิดของตัวแปร array ที่ต้องการ		
ArrayName	ชื่อตัวแปร		
[n]	ขนาดของตัวแปรที่ต้องการ		
ตัวอย่าง	<code>int[] x ;</code> ประกาศ x เป็นตัวแปร array ชนิดจำนวนเต็ม <code>x = new int[10];</code> โดย x เกิดจากการสร้างวัตถุ int หรือ <code>int[] x = new int[10];</code>		

ตัวอย่าง 2.2 TestArray1		Note
<div><div>1</div><div>class TestArray1</div><div>2</div><div>{</div><div>3</div><div> public static void main(String[] args)</div><div>4</div><div> {</div><div>5</div><div> int[] data = {1,2,3,4};</div><div>6</div><div> System.out.println("data[2]: "+data[2]);</div><div>7</div><div> data[2] = data[0] + data[1]*data[2];</div><div>8</div><div> System.out.println("data[2]: "+data[2]);</div><div>9</div><div> }</div><div>10</div><div>}</div></div>	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5 ประกาศตัวแปร array ชื่อ data พร้อมกำหนดค่าเริ่มต้น โดยใช้คำสั่ง</p> <pre>int[] data = {1,2,3,4};</pre> <p>ซึ่งการประกาศตัวแปร array สามารถประกาศแบบนี้ก็ได้ โดยไม่ต้องใช้คำสั่ง new</p>	
<div><div>1</div><div>---- exec: TestArray</div><div>2</div><div></div><div>3</div><div>data[2]: 3</div><div>4</div><div>data[2]: 7</div><div>5</div><div></div><div>6</div><div>---- operation: complete</div></div>		

ตัวอย่าง 2.3 TestArray2		Note
1	class TestArray2	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3 บรรทัดที่ 6 ประกาศตัวแปร array ชื่อ a ในรูปแบบที่ 2 โดยใช้คำสั่ง
2	{	
3	public static void main(String[] args)	
4	{	
5	int sum = 0;	
6	int[] a = new int[5];	

7	for (int i = 0; i < a.length; i++)	int[] a = new int[5];
8	a[i] = 3 + i;	
9	for (int i = 0; i < a.length; i++)	
10	sum += a[i];	
11	System.out.println("Sum: "+sum);	
12	}	
13	}	
1	---- exec: TestArray2	
2		
3	Sum: 25	
4		
5	---- operation: complete	

ตัวอย่าง 2.4 TestArray3		Note
1	class TestArray3	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5 ประกาศตัวแปร array ชื่อ tri แบบ 2 มิติ โดยมีจำนวน 4 rows และความยาวของแต่ละ row ไม่ได้กำหนด โดยใช้คำสั่ง</p> <pre>int[][] tri = new int[4][];</pre> <p>บรรทัดที่ 6-9 กำหนดความยาวแต่ละ row โดย tri[0] มีความยาว 4 และ tri[1], tri[2], tri[3] มีความยาว 3, 2, 1 ตามลำดับ โดยใช้คำสั่ง</p> <pre>tri[0] = new int[4]; tri[1] = new int[3]; tri[2] = new int[2]; tri[3] = new int[1];</pre>
2	{	
3	public static void main(String[] args)	
4	{	
5	int[][] tri = new int[4][];	
6	tri[0] = new int[4];	
7	tri[1] = new int[3];	
8	tri[2] = new int[2];	
9	tri[3] = new int[1];	
10	int n = 1;	
11	for(int i = 0; i < tri.length; i++){	
12	for(int j = 0; j < tri[i].length; j++){	
13	tri[i][j] = n;	
14	n++;	
15	}	
16	}	
17	for(int i = 0; i < tri.length; i++){	
18	for(int j = 0; j < tri[i].length; j++){	
19	System.out.print(tri[i][j] + " ");	
20	}	
21	System.out.println();	
22	}	
23	}	
24	}	
1	---- exec: TestArray3	
2		
3	1 2 3 4	
4	5 6 7	
5	8 9	
6	10	
7		
8	---- operation: complete	

2.3 ตัวดำเนินการ (Operator)

1) ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

เป็นตัวดำเนินการทางคณิตศาสตร์ขั้นพื้นฐาน การเขียนโปรแกรมจะใช้งานตัวดำเนินการชนิดนี้มากที่สุด รายละเอียดของ Arithmetic Operator แสดงดังตารางที่ 2.2

ตารางที่ 2.2 ตารางแสดงตัวดำเนินการทางคณิตศาสตร์พร้อมความหมายและตัวอย่าง

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	การทำงาน
+	การบวก	c = a+b	c ← a+b
-	การลบ	c = a-b	c ← a-b
*	การคูณ	c = a*b	c ← a*b
/	การหาร	c = a/b	c ← a/b
%	การหารเอาเศษ	c = a%b	c ← a%b
++	การเพิ่มค่าขึ้นอีก 1	c = a++	c ← a a ← a+1
		c = ++a	a ← a+1 c ← a
--	การลบค่าลงอีก 1	c = a--	c ← a a ← a-1
		c = --a	a ← a-1 c ← a

ตัวอย่าง 2.5 TestArithmetic1		Note
<pre> 1 class TestArithmetic1 2 { 3 public static void main(String[] args){ 4 int i = 3; 5 i++; 6 System.out.println(i); 7 ++i; 8 System.out.println(i); 9 System.out.println(++i); 10 System.out.println(i++); 11 System.out.println(i); 12 } 13 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int i = 3;</p> <p>บรรทัดที่ 5 คำสั่ง i++;</p> <p>บรรทัดที่ 7 คำสั่ง ++i;</p> <p>บรรทัดที่ 9 คำสั่ง ++i;</p> <p>บรรทัดที่ 10 คำสั่ง i++;</p>
<pre> 1 ---- exec: TestArithmetic1 2 3 4 4 4 5 5 6 6 6 7 7 8 9 ---- operation: complete</pre>		

ตัวอย่าง 2.6 TestArithmetic2		Note
<pre> 1 class TestArithmetic2 2 { 3 public static void main (String[] args)</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p>

4	{	บรรทัดที่ 5 คำสั่ง int result = 1+2;
5	int result = 1 + 2;	บรรทัดที่ 7 คำสั่ง result = result - 1;
6	System.out.println(result);	บรรทัดที่ 9 คำสั่ง result = result * 2;
7	result = result - 1;	บรรทัดที่ 11 คำสั่ง result = result / 2;
8	System.out.println(result);	บรรทัดที่ 13 คำสั่ง result = result + 8;
9	result = result * 2;	บรรทัดที่ 14 คำสั่ง result = result % 7;
10	System.out.println(result);	
11	result = result / 2;	
12	System.out.println(result);	
13	result = result + 8;	
14	result = result % 7;	
15	System.out.println(result);	
16	}	
17	}	
1	---- exec: TestArithmetic2	
2		
3	3	
4	2	
5	4	
6	2	
7	3	
8		
9	---- operation: complete	

2) ตัวดำเนินการกำหนดค่า (Assignment Operators)

เครื่องหมายกำหนดค่าให้กับตัวแปร คือ เครื่องหมายเท่ากับ (=) และภาษา Java สามารถนำ Assignment operator มาทำงานร่วมกับ Arithmetic operation ได้ยกตัวอย่างเช่น $x += y$ มีค่าเท่ากับ $x = x + y$ โดยตัวดำเนินการกำหนดค่า แสดงดังตารางที่ 2.3

ตารางที่ 2.3 ตารางแสดงตัวดำเนินการกำหนดค่า ความหมายและตัวอย่าง

Operator	ความหมาย	ตัวอย่าง	การทำงาน
=	การกำหนดค่า	$x = y$	$x \leftarrow y$
+=	การกำหนดค่าและการบวก	$x += y$	$x \leftarrow x + y$
-=	การกำหนดค่าและการลบ	$x -= y$	$x \leftarrow x - y$
*=	การกำหนดค่าและการคูณ	$x *= y$	$x \leftarrow x * y$
/=	การกำหนดค่าและการหาร	$x /= y$	$x \leftarrow x / y$
%=	การกำหนดค่าและการ modulus	$x \% = y$	$x \leftarrow x \% y$

ตัวอย่าง 2.7 TestAssign1		Note
1	class TestAssign1	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3
2	{	
3	public static void main(String[] args){	

4 5 6 7 8 9 10 11 12 13 14 15 16 17	<pre> int j = 2; System.out.println("j = " + j); j+=4; System.out.println("j add = " + j); j-=4; System.out.println("j sub = " + j); j*=4; System.out.println("j mul = " + j); j/=4; System.out.println("j div = " + j); j%=4; System.out.println("j mol = " + j); } </pre>	บรรทัดที่ 4 คำสั่ง int j = 2; บรรทัดที่ 6 คำสั่ง j += 4; บรรทัดที่ 8 คำสั่ง j -= 4; บรรทัดที่ 10 คำสั่ง j *= 4; บรรทัดที่ 12 คำสั่ง j /= 4; บรรทัดที่ 14 คำสั่ง j %= 4;
1 2 3 4 5 6 7 8 9 10	<pre> ---- exec: TestAssign1 j = 2 j add = 6 j sub = 2 j mul = 8 j div = 2 j mol = 2 ---- operation: complete </pre>	

ตัวอย่าง 2.8 TestAssign2		Note
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	<pre> class TestAssign2 { public static void main(String[] args){ int result =+ 1; System.out.println(result); result--; System.out.println(result); result++; System.out.println(result); result = -result; System.out.println(result); boolean success = false; System.out.println(success); System.out.println(!success); } } </pre>	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3 บรรทัดที่ 4 คำสั่ง int result =+ 1; บรรทัดที่ 6 คำสั่ง result--; บรรทัดที่ 8 คำสั่ง result++; บรรทัดที่ 10 คำสั่ง result = -result; บรรทัดที่ 12 คำสั่ง success = false;
1 2 3 4 5 6 7 8 9 10	<pre> ---- exec: TestAssign2 1 0 1 -1 false true ---- operation: complete </pre>	

3) ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

เครื่องหมายดำเนินการเปรียบเทียบ จะนำมาใช้เมื่อมีการเปรียบเทียบค่าที่เกิดขึ้น โดยผลลัพธ์ที่ได้จากการเปรียบเทียบจะได้ค่า Boolean (True or False) เช่น (9 >= 3) จะได้ค่า True โดยตัวดำเนินการเปรียบเทียบ แสดงดังตารางที่ 2.4

ตารางที่ 2.4 ตารางแสดงตัวดำเนินการเปรียบเทียบ ความหมายและตัวอย่าง

ตัวดำเนินการเปรียบเทียบ	ความหมาย	ตัวอย่าง
==	เท่ากับ	$x == y$
!=	ไม่เท่ากับ	$x != y$
<	น้อยกว่า	$x < y$
<=	น้อยกว่าหรือเท่ากับ	$x <= y$
>	มากกว่า	$x > y$
>=	มากกว่าหรือเท่ากับ	$x >= y$

ตัวอย่าง 2.9 TestComparison1		Note
<pre> 1 class TestComparison1 2 { 3 public static void main(String[] args) 4 { 5 int value1 =1; 6 int value2= 2; 7 if(value1 == value2) 8 System.out.println("val1 == val2"); 9 if(value1 != value2) 10 System.out.println("val1 != val2"); 11 if(value1 > value2) 12 System.out.println("val1 > val2"); 13 if(value1 < value2) 14 System.out.println("val1 < val2"); 15 if(value1 <= value2) 16 System.out.println("val1 <= val2"); 17 } 18 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5 คำสั่ง int value1 = 1;</p> <p>บรรทัดที่ 6 คำสั่ง int value2 = 2;</p> <p>บรรทัดที่ 7 คำสั่ง if(value1==value2)</p> <p>บรรทัดที่ 9 คำสั่ง if(value1 != value2)</p> <p>บรรทัดที่ 11 คำสั่ง if(value1 > value2)</p> <p>บรรทัดที่ 13 คำสั่ง if(value1 < value2)</p> <p>บรรทัดที่ 15 คำสั่ง if(value1<= value2)</p>
<pre> 1 ---- exec: TestComparison1 2 3 val1 != val2 4 val1 < val2 5 val1 <= val2 6 7 ---- operation: complete</pre>		

ตัวอย่าง 2.10 TestComparison2	Note
-------------------------------	------

1	class TestComparison2	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3
2	{	
3	public static void main(String[] args)	
4	{	บรรทัดที่ 5 คำสั่ง int value1 = 1;
5	int value1 = 1;	บรรทัดที่ 6 คำสั่ง int value2 = 3;
6	int value2 = 3;	บรรทัดที่ 7 คำสั่ง
7	if((value1 == 1) && (value2 == 2))	if((value1 == 1) && (value2 == 2))
8	System.out.println(" AND ");	
9	if((value1 == 1) (value2 == 3))	บรรทัดที่ 9 คำสั่ง
10	System.out.println(" OR ");	if((value1 == 1) (value2 == 3))
11	}	
12	}	
1	---- exec: TestComparison2	
2		
3	OR	
4		
5	---- operation: complete	

4) ตัวดำเนินการทางตรรกศาสตร์ (Logical Operators)

ตัวดำเนินการทางตรรกศาสตร์ ใช้สำหรับกำหนดเงื่อนไขเมื่อเงื่อนไขมีมากกว่า 1 เงื่อนไข เครื่องหมาย Logical operators แสดงดังตารางที่ 2.5

ตารางที่ 2.5 ตารางแสดงตัวดำเนินการตรรกศาสตร์ ความหมาย ตัวอย่างและคำอธิบาย

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	คำอธิบาย
&&	logical and	A && B	ผลลัพธ์มีค่าเป็นจริงเมื่อ A และ B มีค่าเป็นจริง
	logical or	A B	ผลลัพธ์มีค่าเป็นจริงเมื่อ A หรือ B มีค่าเป็นจริง
!	not	!A	ผลลัพธ์มีค่าเป็นจริงเมื่อ A มีค่าเป็นเท็จ

ตัวอย่าง 2.11 TestBoolean1		Note
1	class TestBoolean1	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3
2	{	
3	public static void main(String[] args)	
4	{	บรรทัดที่ 5 คำสั่ง int x = 0;
5	int x = 0;	บรรทัดที่ 6 คำสั่ง แสดงผลลัพธ์
6	System.out.println((x != 0)&&((10/x)>1));	((x != 0) && ((10/x)>1))
7	}	
8	}	
1	---- exec: TestBoolean1	
2		
3	false	
4		
5	---- operation: complete	

ตัวอย่าง 2.12 TestBoolean2	Note
----------------------------	------

1	class TestBoolean2	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3
2	{	
3	public static void main(String[] args)	
4	{	บรรทัดที่ 5 คำสั่ง int n = 27;
5	int n = 27;	บรรทัดที่ 6 คำสั่ง
6	boolean even = (n % 2 == 0);	boolean even = (n % 2 == 0);
7	System.out.println(n+" is even? "+ even);	
8	}	
9	}	
1	---- exec: TestBoolean2	
2		
3	27 is even? false	
4		
5	---- operation: complete	

5) ลำดับความสำคัญของตัวดำเนินการ (Operator Precedence)

เมื่อต้องการประมวลผลการดำเนินการมากกว่า 1 ตัวดำเนินการใน 1 expression เช่น $30 + 10/2$ ซึ่งการคำนวณหาค่า x จะสามารถทำได้ 2 วิธี

$$\frac{30+10}{2} = 20 \quad \text{หรือ} \quad 30 + \frac{10}{2} = 35$$

จากทั้ง 2 วิธี อันไหนเป็นคำตอบที่ถูกระหว่าง 20 หรือ 35? นั้นขึ้นอยู่กับว่า compiler จะทำการประมวลผล operation ใดก่อน (การบวก หรือ การหาร) สำหรับ Java ได้กำหนดระดับความสำคัญดังตาราง 2.6 โดยระดับความสำคัญสูงจะถูกประมวลผลก่อน และถ้าระดับความสำคัญเท่ากันก็ให้เรียงตามระดับการกระทำ

ตารางที่ 2.6 แสดงระดับความสำคัญของตัวดำเนินการ (Operator Precedence)

ระดับความสำคัญ	ตัวดำเนินการ	การกระทำ
1	(), []	ซ้ายไปขวา
2	++, --, !, ~	ขวาไปซ้าย
3	*, /, %	ซ้ายไปขวา
4	+, -	ซ้ายไปขวา
5	<<, >>	ซ้ายไปขวา
6	<, <=, >, >=	ซ้ายไปขวา
7	==, !=	ซ้ายไปขวา
8	&	ซ้ายไปขวา
9	^	ซ้ายไปขวา
10		ซ้ายไปขวา
11	&&	ซ้ายไปขวา
12		ซ้ายไปขวา
13	?:	ซ้ายไปขวา
14	=, +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, !=	ขวาไปซ้าย

เมื่อเราทราบระดับความสำคัญตามตารางที่ 2.6 แล้ว $x = 30 + 10/2$ จะประมวลผลตามแบบที่ 2 คือ $30 + 10/2 = 35$ เนื่องจากเครื่องหมายหาร (/) มีระดับความสำคัญสูงกว่าเครื่องหมายบวก (+) โดยลำดับความสำคัญของตัวดำเนินการแสดงได้ดังนี้

ตัวอย่าง 2.13 จงคำนวณหาผลลัพธ์ของแต่ละ Expression ต่อไปนี้

นิพจน์	ขั้นตอนการคิด	ผลลัพธ์
25+16-10	(25+16)-10	31
50-10*3	50-(10*3)	20
70/10*3	(70/10)*3	21
15*2/3	(15*2)/3	10

จากปัญหาของระดับความสำคัญที่กล่าวไปแล้วนั้น ถ้าเราต้องการป้องกันความสับสนให้กับตัวเองนั้น เราสามารถใส่วงเล็บ () เพื่อจัดกลุ่มการคำนวณก่อนหลังได้แสดงดังตาราง Column ที่ 2 ของตัวอย่างที่ 2.13

6) Casting Operator

การประมวลผลในแต่ละ expression เช่น $\text{int } x = 30.2 + 10/2$ ผลลัพธ์ได้เท่ากับ 35.2 ซึ่งเป็นเลขทศนิยมแต่ถูกนำไปเก็บในตัวแปรชนิด int ชื่อ x ดังนั้นจึงเกิดปัญหาความถูกต้องของข้อมูล เมื่อเกิดปัญหาลักษณะนี้ขึ้นภาษา Java ได้กำหนดวิธีการจัดการด้วยการ casting operator ดังนี้

รูปแบบการ Casting Operator

valNameResult = (dataType)VarName;	
dataType	ชนิดของตัวแปร เช่น int, float, double เป็นต้น
VarName	ชื่อตัวแปร
valNameResult	ชื่อตัวแปรที่รับค่าจากการแปลงข้อมูล มีชนิดเดียวกันกับ dataType
การใช้งาน	<ol style="list-style-type: none"> เปลี่ยนจากชนิดข้อมูลที่มีขนาดเล็กไปเป็นชนิดข้อมูลขนาดใหญ่กว่า เช่น <pre>int i = 2530; float f ; f = (float)i;</pre> การเปลี่ยนข้อมูลที่มีขนาดใหญ่กว่าไปเป็นข้อมูลที่มีขนาดเล็กกว่า ซึ่งการแปลงลักษณะนี้จะทำให้ความถูกต้องของข้อมูลลดลง ต้องเพิ่มความระมัดระวัง

ตัวอย่าง 2.14 TestCast1		Note
<pre> 1 class TestCast1 2 { 3 public static void main(String[] args){ 4 int i = 1235; 5 float f = (float)1.3; 6 double d = 257.23; 7 System.out.println(f); 8 System.out.println((float)i); 9 10 f = (float)i; 11 System.out.println(f); 12 i = (int)2.05; 13 System.out.println(i); 14 i = (int)d; 15 System.out.println(i); 16 } 17 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int i = 1235;</p> <p>บรรทัดที่ 5 คำสั่ง float f = (float)1.3;</p> <p>บรรทัดที่ 6 คำสั่ง double d = 257.23;</p> <p>บรรทัดที่ 7 คำสั่ง แสดงผลลัพธ์ f</p> <p>บรรทัดที่ 8 คำสั่ง แสดงผลลัพธ์ (float)i</p> <p>บรรทัดที่ 10 คำสั่ง f = (float)i;</p> <p>บรรทัดที่ 12 คำสั่ง i = (int)2.05;</p> <p>บรรทัดที่ 14 คำสั่ง i = (int)d;</p>
<pre> 1 ---- exec: TestCast1 2 3 1.3 4 1235.0 5 1235.0 6 2 7 257 8 9 ---- operation: complete</pre>		

ตัวอย่าง 2.15 TestCast2		Note
<pre> 1 class TestCast2 2 { 3 public static void main(String[] args) 4 { 5 float x = 1.3f; 6 float y = 1.8f; 7 System.out.println((int)(x+y)); 8 System.out.println((int)x + (int)y); 9 System.out.println(3 + 2); 10 System.out.println("3" + "2"); 11 } 12 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5 คำสั่ง float x = 1.3f;</p> <p>บรรทัดที่ 6 คำสั่ง float y = 1.8f;</p> <p>บรรทัดที่ 7 คำสั่ง แสดงผลลัพธ์ (int)(x+y)</p> <p>บรรทัดที่ 8 คำสั่ง แสดงผลลัพธ์ (int)x + (int)y</p>
<pre> 1 ---- exec: TestCast2 2 3 3 4 2 5 5 6 32 7 8 ---- operation: complete</pre>		

ตัวอย่าง 2.16 TestCast3		Note
<pre> 1 class TestCast3 2 { 3 public static void main(String[] args){ 4 int f = 100; 5 System.out.println(5/9); 6 System.out.println(5*(f-32)/9); 7 System.out.println((5/9)*(f-32)); 8 System.out.println(((float)5/9)*(f-32)); 9 System.out.println((float)5/9); 10 } 11 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int f = 100;</p> <p>บรรทัดที่ 5 คำสั่ง แสดงผลลัพธ์ 5/9</p> <p>บรรทัดที่ 6 คำสั่ง แสดงผลลัพธ์ 5*(f-32)/9</p> <p>บรรทัดที่ 7 คำสั่ง แสดงผลลัพธ์ ((5/9)*(f-32))</p> <p>บรรทัดที่ 8 คำสั่ง แสดงผลลัพธ์ ((float)5/9)*(f-32)</p> <p>บรรทัดที่ 9 คำสั่ง แสดงผลลัพธ์ (float)5/9</p>
<pre> 1 ---- exec: TestCast3 2 3 0 4 37 5 0 6 37.77778 7 0.5555556 8 9 ---- operation: complete</pre>		

2.4 แบบฝึกหัดท้ายบท

- กำหนดให้ int a=1 และ double d=1.0 จงหาผลลัพธ์จาก expressions ต่อไปนี้

ลำดับคำสั่ง	ผลลัพธ์
a=46/9;	
a=46%9+4*4-2;	
a=45+43%5*(23*3%2);	
a %=3/a + 3;	
d=4+d*d+4;	
d+=1.5*3(++a);	
d-=1.5*3+a++;	

- กำหนดให้ int a=5, int b=3, int c=2, int d=0.5 ให้แสดงค่าของ y โดยกำหนดให้ y มีชนิดข้อมูล float

ลำดับคำสั่ง	ผลลัพธ์
y=a*b+c;	
y=b+c*b;	
y=a*a+b*b+c*c;	
y=c%5;	
y=a/c;	
y=a/d	

3. จงเขียนโปรแกรมหาค่านิพจน์ทางคณิตศาสตร์ต่อไปนี้ โดยกำหนดให้รับค่า a,b,c,d เป็นจำนวนเต็มรับค่าจากคำสั่ง scan

นิพจน์ทางคณิตศาสตร์	กำหนดค่า a b c d ตามลำดับ ถ้าไม่มีให้ใส่ค่า 0	ผลลัพธ์
1. $\left(3\frac{a}{5}+\frac{1}{b}\right)$	3 2 0 0	
2. $\left(\frac{3a+5b}{2+c}\right)$	2 8 6 0	
3. $\frac{2}{7}\left(\left(4^{3+c}\right)-5d\right)$	0 0 4 9	
4. $\sqrt{\frac{2+8b}{a}}$	2 6 0 0	
5. $\sqrt[3]{b^2-4d}$	0 3 0 5	

4. เมื่อกำหนดให้ค่าของตัวแปรต่าง ๆ ในหน่วยความจำเป็นดังนี้

	หน่วยความจำ
i	10
j	3
x	1.525
y	-0.008
z	12.26
c	'A'
d	'F'

จงหาผลลัพธ์จากการประมวลผลด้วยตัวเองและการเขียนโปรแกรมหาค่าของนิพจน์ต่อไปนี้

ลำดับนิพจน์ทางคณิตศาสตร์	ผลลัพธ์ประมวลผล ด้วยตนเอง	ผลลัพธ์โปรแกรม
$(y-2)*(y+z)/j$		
$j\%(i-j)/(z-x)$		
$((i/3-1)+((j-1)*6)\%(i-9))*3$		
$-x+(y*y+4*x*z)/x$		
$(c/d)*(-d)$		
$!(c<99)$		
$!(i*j<c)$		
$(c==97)\&\&!(z>15)$		
$(z/2-j<x)\ (i-j!=0)\ (c>d)$		
$(j-i/j)>(d-c/d)$		

โจทย์เพิ่มทักษะการเขียนโปรแกรม

จงเขียนโปรแกรมเพื่อหาผลลัพธ์ของปัญหาข้อ 5 – 10

5. จงเขียนโปรแกรมคำนวณสมการ $3x^4 + 2x^3 - x + 10$ โดย x เป็นจำนวนเต็มที่รับค่าผ่าน keyboard

6. กำหนดให้ [23, 16, 27] จงเขียนโปรแกรมหาค่ามากที่สุดห้ามใช้คำสั่ง for loop

7. กำหนดให้ [23, 58, 88, 16, 27] จงเขียนโปรแกรมหาค่ามากที่สุดห้ามใช้คำสั่ง for loop

8. Write a program that read an integer between 100 and 999, and find the reverse of that integer. For Example, if an integer is 123, its reverse is 321

9. Write the program that prompts the user to enter the seconds (e.g., 86520), and displays the number of days and minutes for the seconds. For Example, if an integer is 86520, 1 days and 2 minutes

10. Suppose you save \$100 each month into a saving account with the annual interest rate 5%. Thus, the monthly interest rate is $0.05/12 = 0.00417$. After the first, second, third months, the value in the account becomes

Month	value
First	$100 \times (1 + 0.00417) = 100.417$
Second	$(100 + 100.417) \times (1 + 0.00417) = 201.252$
Third	$(100 + 201.252) \times (1 + 0.00417) = 302.507$

Write the program that prompts the user to enter a monthly saving amount and display the account value.

บทที่ 3

คำสั่งควบคุม (Control Flow Statement)

3.1 คำสั่งควบคุมแบบมีทางเลือก (Selection Control Statement)

1) If-then statements (If ทางเดียว)

If เป็นคำสั่งที่ใช้ในการตัดสินใจ สมมติว่าลองจินตนาการในชีวิตประจำวัน คนเราจะเกี่ยวข้องกับการใช้คำสั่ง If เสมอตั้งแต่เกิดมาจนถึงปัจจุบันนี้ เช่น ถ้าเราหิวข้าวแล้วเราก็จะกิน ซึ่งสามารถแปลงเป็นภาษา computer ได้ดังนี้

if (😊 == หิว) then { กิน }

รูปแบบ IF ทางเดียว

<pre>if (condition) { Statements; }</pre>	
condition	เงื่อนไข (😊 == หิว) โดยจะต้องบอกได้ว่า มีค่าเป็น true หรือ false เช่น (age >= 30) ผลลัพธ์ true (age = 40), false (age = 20)
Statements	ชุดคำสั่งที่จะทำงานเมื่อ condition เป็น true

ตัวอย่าง 3.1 TestIfOneway		Note
1	class TestIfOneway	โปรแกรมเริ่มทำงานที่บรรทัดที่ 3 บรรทัดที่ 5 คำสั่ง int busPrice = 120; บรรทัดที่ 6 คำสั่ง int money = 100; บรรทัดที่ 7 คำสั่ง if(money < busPrice) busPrice)
2	{	
3	public static void main(String[] args)	
4	{	
5	int busPrice = 120;	
6	int money = 100;	
7	if(money < busPrice)	
8	{	
9	System.out.println("Can not Go");	
10	}	
11	}	
12	}	
1	---- exec: TestIfOneway	
2		
3	Can not Go	
4		
5	---- operation: complete	

2) If-then-else statements (If สองทาง)

จากคำสั่ง If ทางเดียว statements ต่างๆ จะถูกประมวลผลเมื่อ condition ให้ค่าความจริงเป็น true เช่น ถ้าเราหิวข้าวแล้วเราก็กิน แต่ถ้าเราไม่หิวแล้ว (condition เป็น false) เราจะทำอะไร ดังนั้น จึงเป็นที่มาของ If สองทาง โดยสามารถแสดงเป็นภาษา computer ได้ดังนี้

if (😊 == หิว) then { กิน } else {นอน}

รูปแบบ IF สองทาง

<pre> if (condition) { True-statements ; } else { False-statements; } </pre>	
Condition	เงื่อนไข โดยจะต้องบอกได้ว่า มีค่าเป็น true หรือ false
True-statements	ชุดคำสั่งที่จะทำงานเมื่อ condition เป็น true
False-statements	ชุดคำสั่งที่จะทำงานเมื่อ condition เป็น false

ตัวอย่าง 3.2 TestIFTwoway		Note
<pre> 1 class TestIFTwoway 2 { 3 public static void main(String[] args) 4 { 5 int currentMoney = 1000; 6 int needMoney = 1950; 7 if(needMoney < currentMoney) 8 { 9 System.out.println("Stock Money:" + 10 (currentMoney-needMoney)); 11 } 12 else 13 { 14 System.out.println("Money not enough"); 15 } 16 } 17 } </pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5 คำสั่ง int currentMoney = 1000;</p> <p>บรรทัดที่ 6 คำสั่ง int needMoney = 1950;</p> <p>บรรทัดที่ 7-15 คำสั่ง if(needMoney < currentMoney) { } else { }</p>
<pre> 1 ---- exec: TestIFTwoway 2 3 Money not enough 4 5 ---- operation: complete </pre>		

3) If แบบที่ 3 (If หลายทาง)

รูปแบบ IF หลายทาง

<pre> if (condition1) { 1-statements; } else if(condition2) { 2-statements; } else { 3-statements; } </pre>	
condition1, condition2	เงื่อนไข โดยจะต้องบอกได้ว่า มีค่าเป็น true หรือ false ของทางเลือกที่ 1 และทางเลือกที่ 2 ตามลำดับ โดยจำนวนทางเลือกจะมีค่าเท่ากับจำนวน condition+1 เช่น จำนวน condition = 2 จำนวนทางเลือกเท่ากับ 3 (2+1)
1-statements	ชุดคำสั่งที่จะทำงานเมื่อ condition1 เป็น true
2-statements	ชุดคำสั่งที่จะทำงานเมื่อ condition2 เป็น true
3-statements	ชุดคำสั่งที่จะทำงานเมื่อ condition1 และ condition2 เป็น false

ตัวอย่าง 3.3 TestIfMulway		Note
<pre> 1 class TestIfMulway 2 { 3 public static void main(String[] args){ 4 int testscore = 76; 5 char grade; 6 if(testscore >= 90){ 7 grade = 'A'; 8 } 9 else if (testscore >= 80){ 10 grade = 'B'; 11 } 12 else{ 13 grade = 'F'; 14 } 15 System.out.println("Grade = " + grade); 16 } 17 } </pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int testscore = 76;</p> <p>บรรทัดที่ 5 คำสั่ง char grade;</p> <p>บรรทัดที่ 6-14</p> <pre> if(testscore >= 90) { } else if (testscore >= 80) { } else { } </pre>
<pre> 1 ---- exec: TestIfMulway 2 3 Grade = F 4 5 ---- operation: complete </pre>		

4) Switch

Switch จะถูกใช้ในการเปรียบเทียบ expression กับค่าเหตุการณ์ที่เป็นไปได้ โดยถ้า expression มีค่า value ตรงกับ case ใด ประโยคควบคุมภายใต้ case นั้นจะถูกประมวลผล ในแต่ละ case จะมี break เป็นตัวหยุดการทำงานของ case นั้น และถ้าค่า expression ไม่ตรงกับค่า value ใน case ใดเลยให้ program เข้าไปทำงานตามประโยคคำสั่งใน default

รูปแบบ switch

<pre>switch(expression) { case value1: 1-statements; break; case value2: 2-statements; break; default: default-statements; break; }</pre>	
expression	เป็นนิพจน์ที่ต้องตรวจสอบว่าตรงกับ value# ใด เพื่อจะได้ไปทำงานตามชุดคำสั่งภายใต้ value# นั้น โดยจำนวนทางเลือกจะมีค่าเท่ากับจำนวน value+1 เช่น จำนวน value = 2 จำนวนทางเลือกเท่ากับ 3 (2+1)
1-statements	ชุดประโยคคำสั่งที่จะทำงานเมื่อ expression = value1 เป็นจริง
2-statements	ชุดประโยคคำสั่งที่จะทำงานเมื่อ expression = value2 เป็นจริง
default-statements	ชุดประโยคคำสั่งที่จะทำงานเมื่อ expression != value1 หรือ value2
break	break จะถูกใช้เพื่อต้องการออกจาก case หรือ block ปัจจุบัน

ตัวอย่าง 3.4 TestSwitch		Note
<pre>1 class TestSwitch 2 { 3 public static void main(String[] args) { 4 int day = 4; 5 switch (day) { 6 case 1: System.out.println("Sunday"); 7 break; 8 case 2: System.out.println("Monday"); 9 break; 10 case 3: System.out.println("Tuesday"); 11 break; 12 case 4: System.out.println("Wednesday"); 13 break; 14 case 5: System.out.println("Thursday"); 15 break; 16 default: System.out.println("Friday"); 17 break; 18 } 19 } 20 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int day = 4;</p> <p>บรรทัดที่ 6-20 คำสั่ง</p> <pre>switch (day) { case 1: break; case 2: break; case 3: break; case 4: break; case 5: break; default: break; }</pre>
<pre>1 ---- exec: TestSwitch 2 3 Wednesday 4 5 ---- operation: complete</pre>		

3.2 คำสั่งควบคุมแบบทำซ้ำ (Iteration Control Statement)

1) While

While loop จะถูกใช้ในการประมวลผลชุดคำสั่งที่ต้องการทำงานวนไปเรื่อยๆ และจะประมวลผลจนกว่าค่า condition เป็น false หรือ while จะทำการประมวลผลวนรอบไปเรื่อยในขณะที่ค่า condition เป็น true

รูปแบบ Switch

<pre>initiation condition while (condition) { statements; change condition; }</pre>	
initiation condition	ค่าเริ่มต้นที่กำหนดให้แก่ Condition
condition	เงื่อนไข โดยจะต้องบอกได้ว่า มีค่าเป็น true หรือ false
statements	ชุดประโยคคำสั่งที่จะทำงานเมื่อ condition เป็นจริง
change condition	การเปลี่ยนแปลง condition เพื่อเป็นการสั่งทำซ้ำ

ตัวอย่าง 3.5 TestWhile		Note
<pre>1 class TestWhile 2 { 3 public static void main(String[] args){ 4 int y = 0; 5 while(y<5) 6 { 7 System.out.println("y is " + y); 8 ++y; 9 } 10 } 11 }</pre>		โปรแกรมเริ่มทำงานที่บรรทัดที่ 3 บรรทัดที่ 4 คำสั่ง int y = 0; บรรทัดที่ 5-9 คำสั่ง while(y<5) { ++y; คำสั่งเปลี่ยนแปลง condition }
<pre>1 ---- exec: TestWhile 2 3 y is 0 4 y is 1 5 y is 2 6 y is 3 7 y is 4 8 9 ---- operation: complete</pre>		

2) Do while

Do while loop จะมีลักษณะการทำงานเหมือน while loop ยกเว้นแต่ที่ do-while ต้องมีการประมวลผลอย่างน้อย 1 รอบการทำงาน ซึ่งการใช้ while เปรียบเหมือนกับการ check condition ก่อนวนรอบ แต่ในขณะที่การใช้ do-while ต้องมีการทำรอบที่ 1 ก่อนแล้วค่อย check condition

รูปแบบ Do while

Initiation condition do { Statements; Change condition; } while (condition);	
Initiation condition	ค่าเริ่มต้นที่กำหนดให้แก่ Condition
Statements	ชุดประโยคคำสั่งที่จะทำงานในรอบแรกและเมื่อ condition เป็นจริง
Change condition	การเปลี่ยนแปลง condition เพื่อเป็นการสั่งทำซ้ำ
condition	เงื่อนไข โดยจะต้องบอกได้ว่า มีค่าเป็น true หรือ false

ตัวอย่าง 3.6 TestDoWhile		Note
<pre> 1 class TestDoWhile 2 { 3 public static void main(String[] args){ 4 int y = 1; 5 do{ 6 System.out.println("y is " + y); 7 y = y*2; 8 } 9 while(y<10); 10 } 11 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int y = 1;</p> <p>บรรทัดที่ 5-9 คำสั่ง</p> <pre> do { y = y*2; คำสั่งเปลี่ยนแปลง condition } while(y<10)</pre>
<pre> 1 ---- exec: TestDoWhile 2 3 y is 1 4 y is 2 5 y is 4 6 y is 8 7 8 ---- operation: complete</pre>		

3) For

For loop ถูกใช้ในการวนรอบที่มีจำนวนครั้งที่แน่นอน

รูปแบบ Do For

<pre>for(initiation; condition; increasing) { Statements; }</pre>	
initiation	ค่าเริ่มต้นที่กำหนดให้แก่ Condition
condition	เงื่อนไข โดยจะต้องบอกได้ว่า มีค่าเป็น true หรือ false
increasing	การเปลี่ยนแปลง condition เพื่อเป็นการสั่งทำซ้ำ
Statements	ชุดประโยคคำสั่งที่จะทำงานเมื่อ condition เป็นจริง

ตัวอย่าง 3.7 TestFor1		Note
<pre> 1 class TestFor1 2 { 3 public static void main(String[] args){ 4 double interest = 1.05; 5 double bath = 1000; 6 for(int i = 0; i<5;i++) 7 { 8 bath = bath*interest; 9 System.out.println(i + bath); 10 } 11 } 12 }</pre>		<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง double interest = 1.05;</p> <p>บรรทัดที่ 5 คำสั่ง double bath = 10000;</p> <p>บรรทัดที่ 6-10 คำสั่ง</p> <pre>for(int i = 0; i<5;i++) { }</pre>
<pre> 1 ---- exec: TestFor1 2 3 1050.0 4 1103.5 5 1159.625 6 1218.5062500000001 7 1280.2815625000003 8 9 ---- operation: complete</pre>		

ตัวอย่าง 3.8 TestFor2

Note

<pre> 1 class TestFor2 2 { 3 public static void main(String[] args) 4 { 5 int n = 1000; 6 double num = 1.0; 7 double denom = 1.0; 8 double pi = 0.0; 9 for (int i = 0; i <= n; i++) 10 { 11 pi += num/denom; 12 num = -num; 13 denom += 2; 14 } 15 pi = pi * 4; 16 System.out.println("After " + n + " 17 iterations: pi = " + pi); 18 } 19 }</pre>	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 5 คำสั่ง int n = 1000;</p> <p>บรรทัดที่ 6 คำสั่ง double num = 1.0;</p> <p>บรรทัดที่ 7 คำสั่ง double denom = 1.0;</p> <p>บรรทัดที่ 8 คำสั่ง double pi = 0.0;</p> <p>บรรทัดที่ 9-14 คำสั่ง</p> <pre> for(int i = 0; i<=n; i++) { } </pre> <p>บรรทัดที่ 15 คำสั่ง pi = pi * 4;</p>
<pre> 1 ---- exec: TestFor2 2 3 After 1000 iterations: pi = 3.1425916543395442 4 5 ---- operation: complete 6</pre>	

4) Break

Break จะถูกใช้ ก็ต่อเมื่อต้องการออกจาก Loop หรือ Block ปัจจุบันที่กำลังทำงานอยู่ break จะต้องวางอยู่ใน Block ของ switch, while, do-while, หรือ for เท่านั้นจะอยู่นอก block เหมือนประโยคธรรมดาไม่ได้

ตัวอย่าง 3.9 TestBreak1		Note
<pre>1 class TestBreak1 2 { 3 public static void main(String[] args){ 4 for(int i = 0; i < 5; ++i){ 5 System.out.print(i + " "); 6 if(i==3){ 7 break; 8 } 9 } 10 } 11 }</pre>	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4-9 คำสั่ง</p> <pre>for(int i = 0; i<=n; i++) { บรรทัดที่ 6 คำสั่ง if(i==3) บรรทัดที่ 7 คำสั่ง break; }</pre>	
<pre>1 ---- exec: TestBreak1 2 3 0 1 2 3 4 5 ---- operation: complete</pre>		

ตัวอย่าง 3.10 TestBreak2	Note
--------------------------	------

<pre> 1 class TestBreak2 2 { 3 public static void main(String[] args){ 4 int[] arrayOfInts 5 ={32,87,3,589,12,1076,2000,8,622,127}; 6 int searchfor = 12; 7 int i; 8 boolean foundIt = false; 9 for(i = 0; i < arrayOfInts.length; i++){ 10 if (arrayOfInts[i] == searchfor){ 11 foundIt = true; 12 break; 13 } 14 } 15 if(foundIt){ 16 System.out.println("Found "+searchfor + 17 " at index " + i); 18 } 19 else{ 20 System.out.println(searchfor+" not in 21 the array"); 22 } 23 } 24 }</pre>	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3 บรรทัดที่ 4 คำสั่ง int[] arrayOfInts; บรรทัดที่ 6 คำสั่ง int searchfor = 12; บรรทัดที่ 7 คำสั่ง int i; บรรทัดที่ 8 คำสั่ง boolean foundIt = false; บรรทัดที่ 9-14 คำสั่ง for(i = 0; i < arrayOfInts.length; i++) { บรรทัดที่ 10 คำสั่ง if () { บรรทัดที่ 11 คำสั่ง break; } } บรรทัดที่ 15-22 คำสั่ง if(foundIt) { } else { }</p>
<pre> 1 ---- exec: TestBreak1 2 3 Found 12 at index 4 4 5 ---- operation: complete</pre>	

5) Continue

คำสั่ง continue เป็นคำสั่งที่บังคับให้โปรแกรมข้ามไปทำงานรอบถัดไปทันที ไม่ต้องสนใจคำสั่งที่เหลืออยู่ในรูป

ตัวอย่าง 3.11 TestBreak1		Note
<pre>1 class TestContinue 2 { 3 public static void main(String args[]){ 4 int i; 5 for(i = 0; i <= 5; i++){ 6 { 7 if((i == 3) (i == 4)) 8 continue; 9 System.out.print(i + " "); 10 } 11 } 12 }</pre>	<p>โปรแกรมเริ่มทำงานที่บรรทัดที่ 3</p> <p>บรรทัดที่ 4 คำสั่ง int i;</p> <p>บรรทัดที่ 5-10 คำสั่ง</p> <pre>for(i = 0; i <= 5; i++) { บรรทัดที่ 7 คำสั่ง if((i == 3) (i == 4)) บรรทัดที่ 8 คำสั่ง continue; }</pre>	
<pre>1 ---- exec: TestBreak1 2 3 0 1 2 5 4 5 ---- operation: complete</pre>		

3.3 แบบฝึกหัดท้ายบท

1. จงแสดงผลลัพธ์ของโปรแกรมในแต่ละข้อ

ข้อ	Source code	ผลลัพธ์
1	<pre> 1 class Test3311 2 { 3 public static void main(String[] args) { 4 double x = 28.0, r; 5 r = x%5.0; 6 System.out.println(r); 7 } 8 }</pre>	
2	<pre> 1 class Test3312 2 { 3 public static void main(String[] args) { 4 int x=2,y=17,z=11,result=5; 5 result -= 2*z%13+y/3+x; 6 System.out.println(result); 7 } 8 }</pre>	
3	<pre> 1 class Test3313 2 { 3 public static void main(String[] args) 4 { 5 for (int i=10; i>0; i-=3) 6 System.out.print("** "); 7 } 8 }</pre>	
4	<pre> 1 class Test3314 2 { 3 public static void main(String[] args) 4 { 5 for(int i=0; i < 10; i++) 6 { 7 if((3 < i)&&(i<=6)) 8 System.out.print("A "); 9 else 10 System.out.print("B "); 11 } 12 } 13 }</pre>	
5	<pre> 1 class Test3315 2 { 3 public static void main(String[] args) 4 { 5 int i=0; 6 while(i<10) 7 { 8 if((i%2) == 0) 9 System.out.print("A "); 10 else if ((i%3) == 0) 11 System.out.print("B "); 12 else 13 System.out.print("C "); 14 i++; 15 } 16 } 17 }</pre>	
ข้อ	Source code	

6	<pre> 1 class Test3316 2 { 3 public static void main(String[] args) 4 { 5 int i=3, j=20; 6 while(i <= j) 7 { 8 System.out.print(" A"); 9 j-=i; 10 i+=2; 11 } 12 System.out.print("\n j = "+j); 13 } 14 }</pre>	
7	<pre> 1 class Test3317 2 { 3 public static void main(String[] args){ 4 int no = 1,count; 5 do{ 6 System.out.print("n = " + no++); 7 count = no-1; 8 while (count < 4){ 9 System.out.print(" c = "+ ++count); 10 if (count == 4) 11 System.out.print("\n"); 12 } 13 }while(no < 3); 14 } 15 }</pre>	

2. จงแสดงผลลัพธ์ของโปรแกรมในแต่ละข้อ

ข้อ	Source code	ผลลัพธ์
1	<pre> 1 class Test3321 2 { 3 public static void main(String[] args){ 4 int i = 0, x = 0; 5 while(i<20){ 6 if(i<20) 7 x += i; 8 ++i; 9 } 10 System.out.println(x); 11 } 12 }</pre>	
2	<pre> 1 class Test3322 2 { 3 public static void main(String[] args) 4 { 5 int i = 0, x = 0; 6 do { 7 if(i % 5 == 0) 8 x++; 9 ++i; 10 }while(i<20); 11 System.out.println(x); 12 } 13 }</pre>	
ข้อ	Source code	

3	<pre> 1 class Test3323 2 { 3 public static void main(String[] args){ 4 int i = 0, x = 0; 5 for(i=1;i<100;i*=2) 6 { 7 x++; 8 } 9 System.out.println(x); 10 } 11 }</pre>	
4	<pre> 1 class Test3324 2 { 3 public static void main(String[] args){ 4 int i = 0, x = 0; 5 for(i = 1; i < 100; ++i) 6 { 7 if(i%2==1) 8 x+=i; 9 else 10 x=-2; 11 } 12 System.out.println(x); 13 } 14 }</pre>	
5	<pre> 1 class Test3325 2 { 3 public static void main(String[] args){ 4 int i = 0, x = 0; 5 for(i=1;i<100;++i) 6 { 7 if(i%2 == 1) 8 x+=i; 9 else 10 x=-2; 11 break; 12 } 13 System.out.println(x); 14 } 15 }</pre>	
6	<pre> 1 class Test3326 2 { 3 public static void main(String[] args){ 4 int i = 0, x = 0; 5 for(i=1;i<100;++i) 6 { 7 if(i%2 == 1) 8 x+=i; 9 else 10 x=-2; 11 continue; 12 } 13 System.out.println(x); 14 } 15 }</pre>	
ข้อ	Source code	

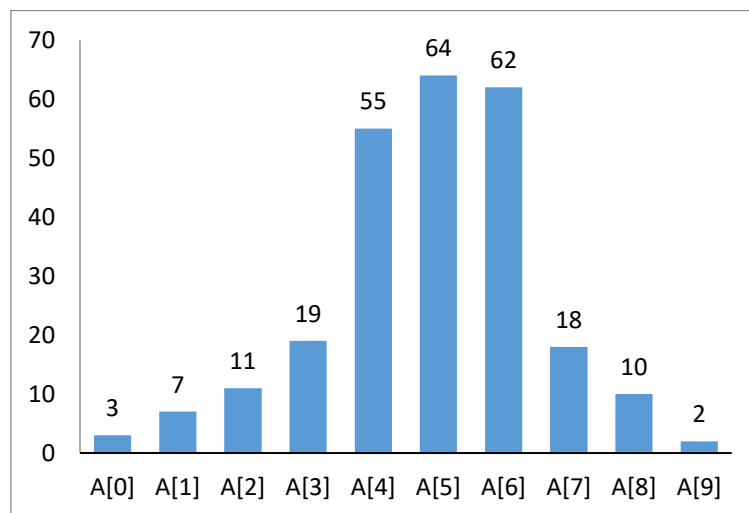
7	<pre> 1 class Test3327 2 { 3 public static void main(String[] args){ 4 int i = 0,j = 0, x = 0; 5 for(i=1; i<10; ++i) 6 for(j=0; j<i; ++j) 7 x += (i+j-1); 8 9 System.out.println(x); 10 } 11 }</pre>	
8	<pre> 1 class Test3328 2 { 3 public static void main(String[] args){ 4 int i = 0,j = 0, x = 0; 5 for(i=1; i<10; ++i) 6 for(j=0; j<i; ++j) 7 { 8 x += (i+j-1); 9 break; 10 } 11 System.out.println(x); 12 } 13 }</pre>	
9	<pre> 1 class Test3329 2 { 3 public static void main(String[] args){ 4 int i,j,k, x = 0; 5 for(i=1; i<5; ++i) 6 for(j=0; j<i; ++j) 7 { 8 k = (i+j-1); 9 x += (i+j-1); 10 if(k%2 == 0) 11 x += k; 12 else 13 if(k%3 == 0) 14 x += k-2; 15 } 16 System.out.println(x); 17 } 18 }</pre>	

โจทย์เพิ่มทักษะการเขียนโปรแกรม

จงเขียนโปรแกรมเพื่อหาผลลัพธ์ของปัญหาข้อ 3 – 12

- เขียนโปรแกรมคำนวณค่าแรง ถ้าทำงาน 40 ชม./สัปดาห์ ได้ค่าแรง ชั่วโมงละ 120 บาท แต่ถ้าทำงานเกิน 40 ชม./สัปดาห์ จะได้ค่า O. T. 20 บาท/ชม.
- เขียนโปรแกรมประมวลผลขาย เครื่องเล่น DVD ทองคำ เมื่อมีผู้ร่วมประมวล 5 คน
- จงเขียนโปรแกรมหาผลรวมที่เป็นเลขคู่ตั้งแต่ 1 ถึง 100
- จงเขียนโปรแกรมคำนวณหาค่า 16 ยกกำลัง 160
- เขียนโปรแกรมบวกและลบเลขใน Matrix ขนาด 3x3

8. เขียนโปรแกรมคูณเลขของ Matrix ขนาด 2×3 กับ 3×2
9. กำหนดให้ $\{25, 2, 8, 16, 35, 4\}$ ให้เลือกมา 2 ตัวและผลรวมมีค่ามากที่สุด
10. กำหนดให้ $\{25, 2, 8, 16, 35, 4\}$ ให้เลือกมา 2 ตัวและผลลบมีค่าน้อยที่สุด
11. กำหนดให้ $\begin{Bmatrix} 25 & 2 & 8 & 16 \\ 10 & 5 & 4 & 3 \end{Bmatrix}$ เป็น matrix ขนาด 2×4 ให้เลือกมา 2 ตัวในแถวที่ 1 ที่ผลรวมมีค่ามากที่สุดที่เป็นไปได้ โดยผลรวมของแถวที่ 2 ที่ column เดียวกับแถวที่ 1 ต้องมีค่าน้อยที่สุด
12. กำหนดให้ $A[n]$ เป็นอาเรย์เก็บข้อมูลจำนวนเต็ม n ตัว ให้นักศึกษาเขียนโปรแกรม $\text{DomeSort}(A[n])$ เรียงลำดับข้อมูล n ตัวให้มีลักษณะแบบ Dome คือ
 - 1) จำนวนคี่ จะถูกเรียงจากน้อยไปมาก ตั้งแต่ $A[0]$ จนถึง $A[k-1]$ เมื่อข้อมูลมีจำนวนคี่ทั้งหมด k ตัว
 - 2) จำนวนคู่ จะถูกเรียงจากมากไปน้อย ต่อจากจำนวนคี่ ตั้งแต่ตำแหน่ง $A[k]$ จนถึง $A[n-1]$
 ดังแสดงในรูป



LAB

6. จงเขียนโปรแกรมอ่านค่าตัวเลขที่อยู่ในช่วง 0-1000 ลงในตัวแปร integer จากนั้นหาผลรวมของตัวเลขดังกล่าว ตัวอย่างเช่น

Input	Output
328	13
400	4
-126	Out of range

7. วันนี้วันที่ 08-07-66

อีก 3 วันคือวันที่ 11-07-66

อีก 10 วันคือวันที่ 18-07-66

แล้วถ้าอีก 123 วัน จะป็นวันอะไร เดือนอะไร ปีอะไร

8. Point in rectangle

จงเขียนโปรแกรมตรวจสอบว่า จุด (x,y) อยู่ในสี่เหลี่ยมมุมฉากที่กำหนดให้หรือไม่

Input บรรทัดที่ 1 คู่ลำดับ x1 y1 จุดเริ่มต้นของสี่เหลี่ยมมุมฉาก มุมล่างซ้าย โดย

$$-1000.0 \leq x1, y1 \leq 1000.0$$

 บรรทัดที่ 2 m n แสดงความกว้าง (m) และความยาว (n) ของสี่เหลี่ยมมุมฉาก
 บรรทัดที่ 3 คู่ลำดับ x0 y0
 Output ถ้าจุด (x0,y0) อยู่ใน สี่เหลี่ยมมุมฉาก ให้ตอบ Yes
 ถ้าจุด (x0,y0) ไม่อยู่ใน สี่เหลี่ยมมุมฉาก ให้ตอบ No

ตัวอย่างผลลัพธ์

Input	Output
0 0 4 4 2 3	Yes

9. Mod dance

มาเต้นกันเถอะ นักศึกษาท่านหนึ่งคิดรหัสการเต้นดังนี้

1. ถ้าจำนวนเต็ม ที่หารด้วย 4 แล้วเหลือเศษ 1 ให้เดินไปข้างหน้า 1 step
2. ถ้าจำนวนเต็ม ที่หารด้วย 4 แล้วเหลือเศษ 2 ให้เดินไปข้างซ้าย 1 step
3. ถ้าจำนวนเต็ม ที่หารด้วย 4 แล้วเหลือเศษ 3 ให้เดินไปข้างขวา 1 step
4. ถ้าจำนวนเต็ม ที่หารด้วย 4 แล้วเหลือเศษ 0 ให้เดินไปข้างหลัง 1 step

ตัวอย่างเช่น ถ้ากำหนดชุดรหัสการเดิน 15 58 69 แล้ว step ในการเดินจะได้ ขวา ซ้าย หน้า เป็นต้น จงเขียนโปรแกรมเดินตามรหัสการเดินและหาจุดสุดท้ายของการเดินว่าตกที่คู่ลำดับ x, y ไตเมื่อกำหนดจุดเริ่มต้นมาให้

Input บรรทัดที่ 1 n โดยที่ $1 \leq n \leq 1000$
 บรรทัดที่ 2 จำนวนข้อมูล n ตัว โดยข้อมูลแต่ละตัวมีค่ามากกว่าเท่ากับศูนย์
 บรรทัดที่ 3 คู่ลำดับ x, y เริ่มต้น โดย x, y เป็นจำนวนเต็มมีค่าอยู่ในช่วง
 $-1000 \leq x, y \leq 1000$

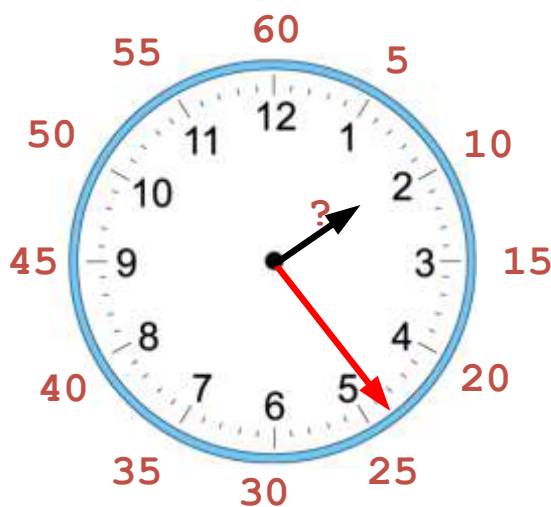
Output คู่ลำดับ x, y จุดสุดท้าย

ตัวอย่างผลลัพธ์

Input	Output
2 12 14 0 0	-1 -1

10. นาฬิกาอนาล็อกเรือนหนึ่งประกอบไปด้วยเลข 1,2,...,59, 60 ที่บอกเวลาเป็นนาที (ตัวเลขด้านนอกวงกลม) และมีตัวเลข 1,2,3,...,12 ที่บอกเวลาเป็นชั่วโมง (ตัวเลขด้านในวงกลม) ถ้าต้องการให้นาฬิกาเรือนนี้บอกเวลา 2.24 น เข็มสั้นจะชี้ที่เลข 12 ของเวลาเป็นนาที

ถ้าเข็มนาฬิกาบอกเวลา 2.48 เข็มสั้นจะชี้ที่เลข 14 ของเวลาเป็นนาที



11. ค่าเฉลี่ย

กำหนดจำนวนเต็ม n จำนวน จงเขียนโปรแกรม หาค่าเฉลี่ยของข้อมูล n จำนวนดังกล่าว โดยให้เขียนอยู่ในรูป function

Input

บรรทัดที่ 1 n แสดงจำนวนเต็ม n จำนวน $1 \leq n \leq 1000000$

บรรทัดที่ 2 i แสดงจำนวนเต็ม n จำนวนเว้นด้วยช่องว่าง $1 \leq i \leq 1000$

Output

บรรทัดที่ 1 ผลลัพธ์ค่าเฉลี่ยของข้อมูล n จำนวน จำนวนทศนิยม 2 ตำแหน่ง

ตัวอย่างผลลัพธ์

Input	Output
2 1 2	1.50
4 5 11 9 1	6.50
3 1 2 3	2.00

12. วางกระเบื้องวงกลม N ชิ้น แต่ละชิ้นมีรัศมีไม่เกิน 10 หน่วยลงบนสนาม โดยกระเบื้องชิ้นที่ i มีจุดศูนย์กลางอยู่ที่พิกัด (x_i, y_i) เป็นเลขจำนวนเต็ม และมีรัศมี r_i เราทราบว่าไม่มีกระเบื้องคู่ใดที่มีจุดศูนย์กลางเป็นจุดเดียวกัน จงเขียนโปรแกรมเพื่อรับตำแหน่งและขนาดของกระเบื้องทั้งหมด จากนั้นคำนวณว่ามีกระเบื้องกี่คู่ที่ซ้อนกัน

ข้อมูลนำเข้า

บรรทัดแรก รับอินพุตเลขจำนวนเต็ม N โดยที่ $1 \leq N \leq 1,000$

N บรรทัดถัดมา เป็นข้อมูลกระเบื้องแต่ละแผ่น กล่าวคือ แต่ละบรรทัด $i + 1$

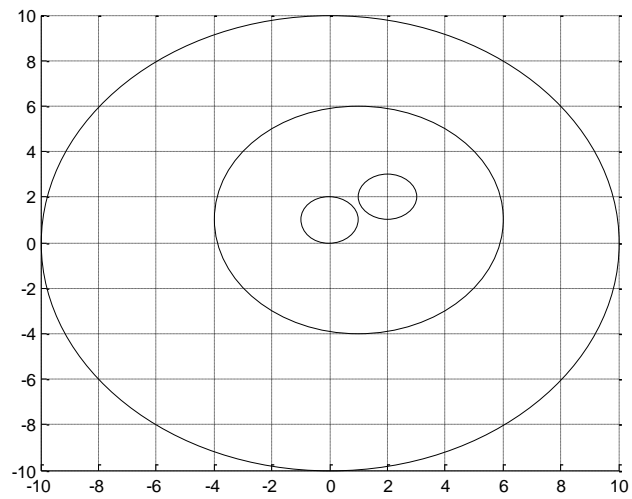
จะรับอินพุตเลขจำนวนเต็ม 3 ค่า ได้แก่ $x_i \ y_i \ r_i$ โดยที่ $-2,000 \leq x_i, y_i \leq 2,000$

และ $1 \leq r_i \leq 10$ รับประกันว่าไม่มีคู่ของ i และ j ที่ $x_i = x_j$ และ $y_i = y_j$ (นั่นคือไม่มีกระเบื้องสองอันใดๆ ที่มีจุดศูนย์กลางอันเดียวกัน)

ข้อมูลส่งออก

แสดงจำนวนคู่ของกระเบื้องที่ซ้อนกัน

ตัวอย่างผลลัพธ์



ข้อมูลนำเข้า	ข้อมูลส่งออก
4 0 0 10 1 1 5 2 2 1 0 1 1	5