មេរៀនទី ៣ : សញ្ញាណនព្វន្តលេខ

១. សញ្ញាណនព្វន្តលេខគណិត

(Arithmetic Operators)

+ − * /

%

++ − −

+= −= *= /= %=

២. សញ្ញាណនព្វន្តលេខគោលពីរ

(The Bitwise Operators)

~	Bitwise unary NOT
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
>>	Shift right
>>>	Shift right zero fill
<<	Shift left
&=	Bitwise AND assignment

|=	Bitwise OR assignment
^=	Bitwise exclusive OR assignment
>>=	Shift right assignment
>>>=	Shift right zero fill assignment
<<=	Shift left assignment

<u>ឧទាហរណ៍</u> : លេខ 42 ដែលមានលំនាំ bit ដូចខាងក្រោម :

00101010

អោយជា 11010101 បន្ទាប់ពីសញ្ញាណពួន្ធលេខគោលពីរ ឈ្នាប់មិន ។

ឧទាហរណ៍ : សញ្ញាណនពួន្ធលេខគោលពីរឈ្នាប់និង "&"

```
  00101010      42
& 00001111      15
-----------------
  00001010      10
```

ឧទាហរណ៍ : សញ្ញាណនពួន្ធលេខគោលពីរឈ្នាប់ឬ "|"

```
  00101010      42
| 00001111      15
-------------------
  00101111      47
```

ឧទាហរណ៍ : សញ្ញាណនព្ធន្ធលេខគោលពីរឈ្នាប់ XOR "^"

```
  00101010      42
^ 00001111      15
----------------
  00100101      37
```

ឧទាហរណ៍ : សញ្ញាណនព្ធន្ធលេខគោលពីរកំលេងៃ, "<<"

រំកិលគ្រប់ bits ទាំងអស់នៅក្នុងតំលៃមួយ ទៅកាន់ខាងឆ្វេង

តាមចំនួន ខ្ទង់ដែលបានកំនត់ ។ វាមានទំរង់ទូទៅ :

value << num

ឧទាហរណ៍ : សញ្ញាណនព្ធន្ធលេខគោលពីរកំណត់ "<<" រំកិលគ្រប់ bits ទាំងអស់នៅក្នុងតំណែលមួយទៅកាន់ខាងស្តាំ តាមចំនួន ខ្ទង់ដែលបានកំនត់ ។ វាមានទំរង់ទូទៅ :

value << num

```
int a = 32;
a = a << 2; // a now contains 8

int a = 35;
a = a << 2; // a still contains 8
```

ឧទាហរណ៍ : សញ្ញាណនព្ធន្តលេខគោលពីរកំលុងស្តាំមិនគិតសញ្ញា, ">>>" ដែលជានិច្ចជាកាលបំពេញលេខ 0 ទៅក្នុង bit លំដាប់ខ្ពស់ ។ លក្ខណៈនេះគេហៅថា unsigned shift ។

```
int a = -1;
a = a >>> 24;
```

ដូចនេះ កំនត់ a ទៅជា 255

នេះជាប្រមាណវិធីដូចគ្នាទៅនឹង code ខាងលើ :

11111111 11111111 11111111 11111111 –1

>>>24

00000000 00000000 00000000 11111111 255

ការកំនត់តំលៃសញ្ញាណពួន្ធលេខគោលពីរ

សញ្ញាណពួន្ធលេខគោលពីរទាំងអស់មានទំរង់អក្សរកាត់ស្រដៀងគ្នាទៅនឹងសញ្ញាណពួន្ធលេខគណិតដែរបើយផ្សំឡើងដោយសញ្ញាណពួន្ធលេខគោលពីរជាមួយនឹងសញ្ញាកំនត់តំលៃ ។

ឧទាហរណ៍ : `a = a >> 4;`

P

ឃ្លានេះសមមូលនឹង :

`a >>= 4;`

៣. សញ្ញាណនព្វន្តលេខធៀប (Relational Operators)

== ស្មើគ្នានឹង

!= មិនស្មើគ្នានឹង

> ធំជាង

< តូចជាង

>= ធំជាង ឬ ស្មើគ្នានឹង

<= តូចជាង ឬ ស្មើគ្នានឹង

លទ្ធផលនៃប្រមាណវិធីជាមួយនឹង
សញ្ញាណនព្វន្តលេខធៀប
អោយជាតំលៃ boolean ។

ឧទាហរណ៍ :

```
int a = 4;
int b = 1;
boolean c = a < b;
```

ករណីនេះ លទ្ធផលនៃ a < b (គឺ false) ត្រូវផ្ទុកទៅក្នុង c ។

```
int done;
// …
if(!done)    …
if(done)     …
```

valid in C/C++ but not in Java.

```
if(done == 0)  …
if(done != 0)  …
```

This is Java-style.

៤. សញ្ញាណនព្វន្តលេខប៉ះបកក្រុងគេន Boolean

(Boolean Logical Operators)

&	Logical AND
|	Logical OR
^	Logical XOR (exclusive OR)
||	Short-circuit OR
&&	Short-circuit AND
!	Logical unary NOT
&=	AND assignment
|=	OR assignment

^= XOR assignment

== Equal to

!= Not equal to

?: Ternary if-then-else

សញ្ញាណនព្ធន្ធលេខបែបតក្កប្រភេទ boolean គឺ &, | និង ^ ធ្វើប្រមាណវិធីអោយតំលៃ boolean ដូចគ្នានឹងប្រមាណវិធីលើ bits នៃចំនួនគត់ដែរ។ សញ្ញាណនព្ធន្ធលេខបែបតក្ក ! បញ្ជ្រាសសភាព boolean : !true == false

និង !false == true

៥. សញ្ញាណនពួកលេខកំណត់តំលៃ

ឧទាហរណ៍ :

```
int x, y, z;
x = y = z = 100;    // set x, y, and z to 100
```

៦. សញ្ញាណនពួកលេខ ? :

រាមានទំរង់ទូទៅ :

expression1 **?** expression2 **:** expression3

P

៧. ការប្រើសញ្ញាវង់ក្រចក

សញ្ញាវង់ក្រចកលើកកំពស់អទិភាពនៃសញ្ញាណនព្វន្តលេខ ដែលស្ថិតនៅក្នុងវា។

```
a >> b + 3
a >>  (b + 3)
(a >> b) + 3
```

អាទិភាពខ្ពស់បំផុត

()	[]	.	
++	--	~	!
*	/	%	
+	-		
>>	>>>	<<	
>	>=	<	<=
==	!=		
&			
^			
|			
&&			
||			
?:			
=	op =		

អាទិភាពនាមបំផុត

តារាងអាទិភាពសញ្ញាណនព្វន្តលេខ

៨. ការប្រើ Methods មួយចំនួនរបស់ Math class

Math class មាន methods ជាច្រើនដែលប្រើតំលៃទសភាគហើយត្រូវបានប្រើនៅក្នុងធរណីមាត្រ និងត្រីកោណមាត្រដូចទៅនឹង methods ទូទៅដែរ។ វាមានតំលៃថេរពីរជាប្រភេទ double គឺ : E (តំលៃ ប្រហែល 2.72) និង PI (តំលៃប្រហែល 3.14)។ Methods សំខាន់ៗមានដូចខាងក្រោម :

static int abs(int *arg*) អោយតំលៃដាច់ខាតនៃ *arg*

static long abs(long *arg*) អោយតំលៃដាច់ខាតនៃ *arg*

static float abs(float *arg*) អោយតំលៃដាច់ខាតនៃ *arg*

static double abs(double *arg*) អោយតំលៃដាច់ខាតនៃ *arg*

static double ceil(double *arg*) អោយតំលៃគត់ដោយធំជាង ឬស្មើនឹង *arg*

static double floor(double *arg*) អោយតំលៃគត់ដោយតូចជាងឬស្មើនឹង *arg*

static int max(int *x*, int *y*) អោយតំលៃអតិបរមានៃ *x* និង *y*

static long max(long *x*, long *y*) អោយតំលៃអតិបរមានៃ *x* និង *y*

static float max(float *x*, float *y*) អោយតំលៃអតិបរមានៃ *x* និង *y*

static double max(double *x*, double *y*) អោយតំលៃអតិបរមានៃ *x* និង *y*

static int min(int *x*, int *y*) អោយតំលៃអប្បបរមានៃ *x* និង *y*

static long min(long *x*, long *y*) អោយតំលៃអប្បបរមានៃ *x* និង *y*

static float min(float *x*, float *y*) អោយតំលៃអប្បបរមានៃ *x* និង *y*

static double min(double *x*, double *y*) អោយតំលៃ

អប្បបរមានៃ *x* និង *y*

static double rint(double *arg*) អោយតំលៃជាលេខគត់

ដែលមានតំលៃក្បែរនឹង *arg*

static int round(float *arg*) អោយតំលៃ *arg* ជាលេខ

គត់ដែលនៅជិតតំលៃរបស់វាប្រភេទ int

static long round(double *arg*) អោយតំលៃ *arg* ជា

លេខគត់ដែលនៅជិតតំលៃរបស់វាប្រភេទ long

static float min(float *x*, float *y*) អោយតំលៃអប្បបរមានៃ *x* និង y

static float signum(float *arg*) អោយតំលៃជាចំនួនមួយតាងអោយ

សញ្ញានៃ *arg* : វាអោយតំលៃ –1.0 កាលណា *arg* មានតំលៃអវិជ្ជមាន

និងអោយតំលៃ 0.0 កាលណា *arg* មានតំលៃសូន្យ ហើយវាអោយតំលៃ

1.0 កាលណា *arg* មានតំលៃវិជ្ជមានប្រភេទ float ។

static double signum(double *arg*) អោយតំលៃជាចំនួនមួយតាង

អោយសញ្ញានៃ arg : វាអោយតំលៃ –1.0 កាលណា *arg* មានតំលៃ

អវិជ្ជមាននិងអោយតំលៃ 0.0 កាលណា *arg* មានតំលៃសូន្យ ហើយវា

អោយតំលៃ 1.0 កាលណា *arg* មានតំលៃវិជ្ជមានប្រភេទ float ។

static double exp(double *arg*) អោយតំលៃជា e ស្វ័យគុណ *arg*

static double log(double *arg*) អោយតំលៃជាឡូការីត *e* នៃ *arg*

static double log10(double *arg*) អោយតំលៃជាឡូការីត គោលដប់នៃ *arg*

static double pow(double *x*, double *y*) អោយ តំលៃជា *x* ស្វ័យគុណ *y*

static double sqrt(double *arg*) អោយតំលៃជាឫសការេនៃ *arg*

static double cbrt(double *arg*) អោយតំលៃជាឫស cube នៃ *arg*

P

static double hypot(double *arg1*, double *arg2*) អោយ
តំលៃជាអ៊ីប៉ូតេនុសនៃត្រីកោណកែងតាមទ្រឹស្តីបទពីតាហ្គ័រ

static double sin(double *arg*) អោយតំលៃស៊ីនុសនៃមុំដែល
កំណត់ដោយ *arg* គិតជារ៉ាដ្យង់

static double cos(double *arg*) អោយតំលៃកូស៊ីនុសនៃមុំដែល
កំណត់ដោយ *arg* គិតជារ៉ាដ្យង់

static double tan(double *arg*) អោយតំលៃតង់ស្យង់នៃមុំដែល
កំណត់ដោយ *arg* គិតជារ៉ាដ្យង់

static double asin(double *arg*) អោយតំលៃជាមុំដែលស៊ីនុស
របស់វាបានកំណត់ដោយ *arg*

P

static double acos(double *arg*) អោយតំលៃជាមុំដែលកូស៊ីនុសរបស់វាបានកំណត់ដោយ *arg*

static double atan(double *arg*) អោយតំលៃជាមុំដែលតង់ស្យង់របស់វាបានកំណត់ដោយ *arg*

static double atan2(double *x*, double *y*) អោយតំលៃជាមុំដែលតង់ស្យង់របស់វាស្មើនឹង x/y

static double random() អោយតំលៃមួយមិនតាមលំដាប់លំដោយ។ តំលៃរបស់វាគឺចន្លោះ 0 និង 1 ។

static double toRadians(double *angle*) អោយ

តំលៃមួយដែលបំលែងពីដឺក្រេទៅជារ៉ាដ្យង់ ។

static double toDegrees(double *radians*) អោយ

តំលៃមួយដែលបំលែងពីរ៉ាដ្យង់មកជាដឺក្រេ

កម្មវិធីខាងក្រោមនេះបង្ហាញនូវការប្រើ toRadians() និង toDegrees() ។

៩. របៀបប្រើ printf() method

ដោយសារ printf() នៅក្នុងភាសា C មានលក្ខណៈងាយស្រួលប្រើនិងមានការនិយមច្រើន ហេតុនេះ Java 1.5 ក៏បានបញ្ចូលនូវ printf() methods សំរាប់ប្រើបង្ហាញទំរង់ output មកលើអេក្រង់ដូចទៅនឹងភាសា C ដែរ។ វាមាន conversion specifiers មួយចំនួនដូចខាងក្រោម :

តួអក្សរ	អត្ថន័យ	ទ្រនាមារណ៍	argument	លទ្ធផល
s	ទំរង់ជា string	%s	"boy"	boy
d	ទំរង់ជាចំនួនគត់(លេខគោលដប់)	%d	myAge	89
f	ទំរង់ជាចំនួនទសភាគ(តាមធម្មតា 6 ខ្ទង់ក្រោយក្បៀស)	%f	Math.PI	3.141593
g	ទំរង់ជាចំនួនទសភាគ(តាមទំរង់ scientific notation)	%g	1000000000	1.000000e+09
x	ទំរង់ជាចំនួនគត់(លេខគោល16)	%x	(int) 'A'	41
b	ទំរង់ជា boolean	%b	5<10	true
c	ទំរង់ជាតួអក្សរ(character)	%c	'A'	A

ឧទាហរណ៍ : គេមានតំលៃលេខទសភាគ គឺ 3.14159 ហើយ

ចង់បង្ហាញលេខនោះដែលនៅខាងក្រោយក្បៀសរបស់វាតែពីរខ្ទង់

លេខប៉ុណ្ណោះ គេអាចសរសេរ :

```
double pi = 3.14159;
System.out.printf("%.2f\n",pi);
```

ករណីបង្ហាញតំលៃលេខក្រោមទំរង់ scientific notation:

```
double val = 123456.7890;
System.out.printf("%.2g\n",val);
System.out.printf("%,f\n",val);
```

១០.ការប្រើ Methods មួយចំនួនរបស់ Scanner class

ដើម្បីអានតំលៃបញ្ចូលជាចំនួនលេខ តួអក្សរ មួយពាក្យ ឬ អក្សរមួយឃ្លាតាមរយៈ keyboard គេអាចប្រើ methods មួយចំនួនរបស់ Scanner class ។ Scanner class បានផ្ទុកនៅក្នុង package មួយឈ្មោះ java.util ។ methods របស់វាមានឈ្មោះដូចខាងក្រោម ៖

boolean nextBoolean() អានតំលៃប្រភេទ boolean

byte nextByte() អានតំលៃប្រភេទ byte

double nextDouble() អានតំលៃប្រភេទ double

float nextFloat() អានតំលៃប្រភេទ float

int nextInt() អានតំលៃប្រភេទ int

String next () អានតំលៃប្រភេទ string (មួយពាក្យ)

String nextLine () អានតំលៃប្រភេទ string (មួយឃ្លា)

long nextLong() អានតំលៃប្រភេទ long

short nextShort() អានតំលៃប្រភេទ short

char findInLine(“.”).charAt(0) អានតំលៃប្រភេទ char

ដើម្បីការពារនូវការអានបញ្ចូលប្រភេទទិន្នន័យអោយបានត្រឹមត្រូវហើយជៀសវាង error នោះគេអាចប្រើ method:

boolean hasNextBoolean() អោយតំលៃ true បើសិនជា តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ boolean

boolean hasNextByte() អោយតំលៃ true បើសិនជា តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ byte

boolean hasNextDouble() អោយតំលៃ true បើសិនជាតំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ double

boolean hasNextFloat() អោយតំលៃ true បើសិនជា

តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ float ។

boolean hasNextInt() អោយតំលៃ true បើសិនជា

តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ int ។

boolean hasNextLong() អោយតំលៃ true បើសិនជា

តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ long ។

boolean hasNextShort() អោយតំលៃ true បើសិនជា

តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាប្រភេទ short ។

boolean hasNext () អោយតំលៃ true បើសិនជា

តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាអក្សរមួយពាក្យ (word) ។

boolean hasNextLine () អោយតំលៃ true បើសិនជា

តំលៃដែលត្រូវបញ្ចូលនៅបន្ទាប់ជាអក្សរមួយបន្ទាត់ (line) ។

សំណួរ និងលំហាត់

១ - តើសញ្ញាណពួន្ធលេខ % ប្រើសំរាប់ធ្វើអ្វី? ហើយវាអាចប្រើបានជាមួយនឹងប្រភេទទិន្នន័យណាខ្លះ?

២ - តើគេអាចប្រើ shift operators សំរាប់គុណ ឬចែកចំនួនគត់មួយទៅនឹងលេខពីរបានដែរឬទេ?

៣ - តើប្រភេទទិន្នន័យណាខ្លះដែលសញ្ញាណពួន្ធលេខគោលពីរ (bitwise operators) អាចប្រើបាន?

៤ - ចូរសរសេរ code ខាងក្រោមនេះទៅជាទំរង់នៃការប្រើ ? operator :

```
if (x < 0)    y = 10;
else y = 20;
```

៥ - សញ្ញា & នៅក្នុង code ខាងក្រោមនេះ តើគេប្រើ bitwise operator ឬ logical operator? ព្រោះអ្វី?

```
boolean a, b;
// ...
if(a & b) ...
```

៦ - ចូរបង្ហាញឈ្នាប់និង (AND) អាចការពារមិនអោយចែកចំនួនមួយទៅនឹងសូន្យបាន ។

៧ - តើ code ខាងក្រោមនេះអោយលទ្ធផលអ្វី?

```
System.out.printf("%(d \n", -23);
System.out.printf("**%2d**  \n", 0);
System.out.printf("%.4s \n", "lengthy");
```

៨ - ចូរសរសេរកម្មវិធីមួយដែលអាចអោយគេបញ្ចូលតំលៃជាអត្តលេខ ឈ្មោះ ភេទ ពិន្ទុគណិត រូប គីមី កុំព្យូទ័ររបស់និស្សិតម្នាក់ពី keyboard រួចបង្ហាញពត៌មានមកលើអេក្រង់តាមទំរង់ខាងក្រោម :

Id Name Sex Total Scores Average Scores