

Reading input from the console

1- create a Scanner object [Scanner input = new Scanner(System.in)

2- Use the method (ex: nextDouble()) to obtain to (double) value [double = input.nextDouble ();]

Reading numbers from the keyboard

TABLE 2.2 Methods for Scanner Objects

Method	Description
nextByte()	reads an integer of the byte type.
nextShort()	reads an integer of the short type.
nextInt()	reads an integer of the int type.
nextLong()	reads an integer of the long type.
nextFloat()	reads a number of the float type.
nextDouble()	reads a number of the double type.

Import

Implicit import (الضمني)

java.util.*;

Explicit import (الصريح)

java.util.Scanner;

Identifiers (شروط التسمية)

1- An identifier is a sequence of characters that consist of letters, digits, underscore _ , dollar sign\$

2- must start with a letter_or \$, it can't with digits

3- can't be a reserved word (Java keywords)

4- can't be true, false, null

5- can be of any length

Variables

1- declaring[ال تصريح]	هو تعريف المتغير مع تحديد نوعه دون إعطائه قيمة	int x;
2- Assignment[الإسناد]	هو إعطاء قيمة للمتغير بعد التصريح عنه	x = 3;
3- declaring and initialisation[التصريح و التهيئة]	هو تعريف المتغير وإعطائه قيمة ابتدائية في نفس السطر	int x = 3;

Naming Conventions

Choose meaningful and descriptive names

1 - variables and method names:

Use lowercase, ex variable : area , ex method : computeArea

2- Class names:

Capitalise the first letter of each word in the name, ex : ComputeArea



Naming Conventions (cont)

3 - constants names:

- Capitalise all letters, ex : PI
- use underscore to separate words, ex : MAX_VALUE

Variables and method names :

If the name consist of several words :

- 1- concatenate all into one
- 2- use lowercase for the first word
- 3- capitalise the first letter of each subsequent word

Numerical Data Types

TABLE 2.1 Numeric Data Types

Name	Range	Storage Size	
byte	-2^7 to $2^7 - 1$ (-128 to 127)	8-bit signed	byte type
short	-2^{15} to $2^{15} - 1$ (-32768 to 32767)	16-bit signed	short type
int	-2^{31} to $2^{31} - 1$ (-2147483648 to 2147483647)	32-bit signed	int type
long	-2^{63} to $2^{63} - 1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed	long type
float	Negative range: $-3.4028235E + 38$ to $-1.4E - 45$ Positive range: $1.4E - 45$ to $3.4028235E + 38$	32-bit IEEE 754	float type
double	Negative range: $-1.7976931348623157E + 308$ to $-4.9E - 324$ Positive range: $4.9E - 324$ to $1.7976931348623157E + 308$	64-bit IEEE 754	double type

Numeric Operators

TABLE 2.3 Numeric Operators

Name	Meaning	Example	Result
+	Addition	$34 + 1$	35
-	Subtraction	$34.0 - 0.1$	33.9
*	Multiplication	$300 * 30$	9000
/	Division	$1.0 / 2.0$	0.5
%	Remainder	$20 \% 3$	2

$5/2$ yields 2 $\rightarrow$ int / int = int

$5.0 / 2$ yields 2.5 $\rightarrow$ double / int = double

$5\%2$ yields 1 باقي القسمة

Remainder operator is very useful in programming :

even%2 = 0

odd %2 =1

Number literals

Integer literals :

An integer literals is assumed to be of the **int** type

To denote an integer literal of the **long** type appond it with **L** or **l**

double literals :

Floating point literal is treated as a **double** type value

1- making a number a **float**

Add **F** or **f**

2- making a number a **double**

Add **D** or **d**

Note :

Complication error would occur if literals were to large for the variable to hold

ex : byte b = 1000;

The **double** type value are **more accurate** then **float** type value



By **Lixxiu**
cheatography.com/lixxiu/

Not published yet.
Last updated 13th December, 2024.
Page 2 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>

Arithmetic Expressions

$$\frac{3+4x}{5} - \frac{10(y-5)(a+b+c)}{x} + 9\left(\frac{4}{x} + \frac{9+x}{y}\right)$$

is translated to

$$(3+4*x)/5 - 10*(y-5)*(a+b+c)/x + 9*(4/x + (9+x)/y)$$

احول الصيغة اللي فوق الى صيغة تفهمها الجافا
الضرب لازم يكون بعلامة*

Increment and Decrement

Table 2.3 Increment and Decrement Operators

Operator	Name	Description	Example: variable = 12
++	pre-increment	Increment variable by 1, and use the new value in the expression.	int i = 12; i++; // i is now 13
++	post-increment	Increment variable by 1, but use the original value in the expression.	int i = 12; i++; // i is now 13
--	pre-decrement	Decrement variable by 1, and use the new value in the expression.	int i = 12; i--; // i is now 11
--	post-decrement	Decrement variable by 1, but use the original value in the expression.	int i = 12; i--; // i is now 11

Augmented Assignment operators

Table 2.4 Augmented Assignment Operators

Operator	Name	Example	Equivalent
+=	Additive assignment	i += 8	i = i + 8
-=	Subtractive assignment	i -= 8	i = i - 8
*=	Multiplicative assignment	i *= 8	i = i * 8
/=	Division assignment	i /= 8	i = i / 8
%=	Remainder assignment	i %= 8	i = i % 8

Conversion Rules قواعد التحويل

If one of the operands is double, the other is converted into double.

Otherwise, if one of the operands is float, the other is converted into float

Otherwise, if one of the operands is long, the other is converted into long.

Otherwise, both operands are converted into int.

Implicit casting

Implicit casting

ex: int x = 10; —> double y = x;

Explicit casting

ex : int x = (int)10.0;

C

By Lixxiu
cheatography.com/lixxiu/

Not published yet.
Last updated 13th December, 2024.
Page 3 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>