

Operators

(1) Arithmetic

:Used in mathematical expressions in the same way that they are used in algebra.

Operator	Result
+	addition
-	subtraction
*	Multiplication
/	Division
%	modulus
++	Increment
--	decrement
+=	Addition assignment
-=	Subtraction assignment

(2) Relational

:they determine the relationship that one operand has to another

Operator	Result
==	equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to

(3) Bitwise

:bitwise operators can be applied to integer types: long,int,short,char and byte

Operator	Result
~	Bitwise unary NOT
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
>>	Shift right
<<	Shift left
&=	Bitwise AND assignment

(4) Logical

:The Boolean logical operators operates only on boolean operands.

All of the binary logical operators combine two boolean values to form a boolean result value.

Operator	Result
&	Logical AND
	Logical OR
^	Logical exclusive OR
	Short-circuit OR
&&	Short-circuit AND
!	Logical unary NOT
&=	AND assignment

(5) Assignment

:The assignment operator is a single equal sign, = ,

The assignment operator works in Java much as it does in any other computer language

(6) Conditional

:Less than, Greater than, Less than or equal to, Greater than or equal to, Not equal to

2. Operator Precedence

Highest						
++(postfix)	++(postfix)					
--(prefix)	--(prefix)	~	!	+(unary)	-(unary)	(type-cast)
*	/	%				
+	-					
>>	>>>	<<				
>	>=	<	<=	instanceof		
==	!=					
&						
^						
&&						
?:						
->						
=	op=					
Lowest						

3. Control Statement

1. Selection

: These statements allow you to control the flow of your program's execution based upon

conditions known only during runtime.

- (1) if
- (2) Nested ifs
- (3) if-else-if ladder
- (4) switch
- (5) Nested switch

2. Iteration

: These statements create loops that repeatedly executes the same set of instructions until a termination condition is met.

- (1) for
- (2) while
- (3) do-while

3. Jump

: These statements transfer control to another part of your program

- (1) break
- (2) continue
- (3) return