Operators

JAVA language supports a rich set of built-in operators. An operator is a symbol that tells the compiler to perform certain mathematical or logical manipulations. Operators are used in program to manipulate data and variables. JAVA operators are binary, unary and ternary operators. A unary operator required one operand, and binary operator required two operands between the one operators. List of main types of JAVA operators. JAVA operators can be classified into following operation types,

S.No.	Operator Type	Description
1	Arithmetic	Performs mathematical calculations.
2	Increment/Decrement	One-by one-increment or decrement arithmetic operations.
3	Comparison/Relational	Compares operands and return a Boolean value
4	Logical	Performs Boolean operations on Boolean operands
5	Assignment	Assign values to variables.
6	Special	Performs various task special things
7	Conditional	Conditional operator is use to apply if -- else condition in single line form and result produced in Boolean true or false.
8	Bitwise	Bitwise operators perform manipulations of data at **bit level**.
9	Array	Performs Operator on arrays.

1. **Arithmetic operators:** JAVA supports all the basic arithmetic operators. The following table shows all the basic arithmetic operators.

Symbol	Description
+	adds two operands
-	subtract second operands from first
*	multiply two operand
/	divide numerator by denominator
%	remainder of division

2. Increment/Decrement operators: JAVA supports all the basic basic Increment or decrement arithmetic operators. The following table shows.

Symbol	Description
++	Increment operator increases integer value by one
--	Decrement operator decreases integer value by one

3. Relation operators: The following table shows all relation operators.

Operator	Description
==	Check if two operand are equal
!=	Check if two operands are not equal.
===	Return true if the operands are Strict equal and of the same data type.
>	Check if operand on the left is greater than operand on the right
<	Check operand on the left is smaller than right operand
>=	check left operand is greater than or equal to right operand
<=	Check if operand on left is smaller than or equal to right operand

4. Logical operators: JAVA language supports following 3 logical operators. Suppose a=1 and b=0,

Operator	Description	Example
&&	Logical AND	(a && b) is false
||	Logical OR	(a || b) is true
!	Logical NOT	(!a) is false

5. Assignment Operators: Assignment operators supported by JAVA.

Operator	Description	Example
=	assigns values from right side operands to left side operand	a=b
+=	adds right operand to the left operand and assign the result to left	a+=b is same as a=a+b
-=	subtracts right operand from the left operand and assign the result to left operand	a-=b is same as a=a-b
*=	multiply left operand with the right operand and assign the result to left operand	a*=b is same as a=a*b
/=	divides left operand with the right operand and assign the result to left operand	a/=b is same as a=a/b
%=	calculate modulus using two operands and assign the result to left operand	a%=b is same as a=a%b

6. Special operator:

Operator	Description
instanceof	Returns true if an object is of a specified object type. This operator is used only for object reference variables. The operator checks whether the object is of a particular type (class type or interface type).
[]	Accesses an element of an array
=>	Specified the index or key of an array element
+	Concatenation operator Combine or attached two or more different type of expression interprets and is put together in a single statement.

7. **Bitwise operators:** Bitwise operators perform manipulations of data at **bit level**. These operators also perform **shifting of bits** from right to left. Bitwise operators are not applied to **float** or **double,** only apply Integer value.

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR(XOR)
<<	left shift
>>	right shift

Truth table for bitwise

X	Y	X & Y	X | Y	X ^ Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

The bitwise shift operators shift the bit value. The left operand specifies the value to be shifted and the right operand specifies the number of positions that the bits in the value are to be shifted. Both operands have the same precedence.

8. **Conditional operators:** It is also known as ternary operator and used to evaluate conditional expression.

Operator	Description
( )? :	Executes one of two expression based on the results of a conditional expression.
(Expression1)?Expe2 : Expre3;	If **Expression1 Condition** is true? value **Expre2**: Otherwise value **Expre3;** Then

9. **Array operators:** Declare the array use array operator. It is also known array initialization. Array operator we discus in chapter array.

Preceding of operator

Operator precedence determines the order in which operators are evaluated. Operators with higher precedence are evaluated first. Operator precedence refers to the order in which operators execute within an expression.

The precedence of an operator specifies how "tightly" it binds two expressions together. For example, in the expression 1 + 5 * 3, the answer is 16 and not 18 because the multiplication ("*") operator has a higher precedence than the addition ("+") operator. Parentheses may be used to force precedence, if necessary. For instance: (1 + 5) * 3 evaluates to 18.

Use of parentheses, even when not strictly necessary, can often increase readability of the code by making grouping explicit rather than relying on the implicit operator precedence and associativity. The following table lists the JAVA operators, ordered from highest to lowest precedence.

Operator (symbols)	Type Operator	Associativity
++ --	Increment and decrement	Right to left
!	Logical Not	Right to Left
/ * %	Multiply , division and modulus	Left to right
+ - .	Addition , subtraction and concatenation string	Left to Right
== <= >= > < != <>	Relational or comparisons	None
&& ||	Logical And , Or	Left to Right
= += -= *= /= %=	Assignment operator	Right to left

Expressions Builds

An **expression** in a programming language is a combination of one or more constants, variables, operators, and functions that the programming language interprets and is put together in a single statement. Variable and data become most useful when you use them in an expression. An expression is a literal value or variable and (or a combination of literal values, variables, operators, and other expressions that can be evaluated by the JAVA scripting engine to produce a result. You use operands and operators to create expression in JAVA.

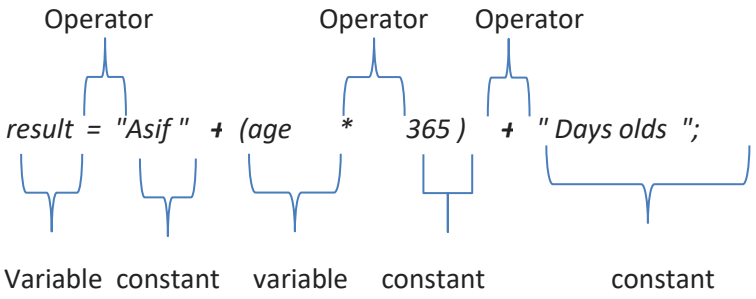
```
package operator.expression;
public class OperatorExpression
{

    public static void main(String[] args)
  {

     int age = 25;
    String result = "Asif  " + (age * 365 ) + " Days olds "; // this is  Expression  for store in  result variable
     System.out.println(result);

  }

}
```



Exercise

Theory Questions

1. What is Operator and describe of relational operators with symbols.
2. What difference between the binary and unary operators?
3. What do you mean by preceding Operators?
4. What do you mean by BODMAS operation?
5. What difference between the (=, ==) operators?

Practical Questions

1. Write a simple program to calculate total numbers of days you lived up to until now

   Hint: Input your age (in years) and then calculate number of days

   number _of _days = ***age*** (in years) * 365

2. Write a code to calculate and print percentage of a student where, Total***_Marks***=400, ***Obtained_Marks***=Sum of obtained marks of all 4 subjects ***Percentage***= (Obtained_Marks*100)/ ***Total_Marks***.

   (Hint: input marks of four subjects out of 100).

3. Write a code to input Temperature in Centigrade and convert to Fahrenheit.
4. If A= 34 and B=55 then solve the following Exercise; Also Justify your answer

   (as example System.out.println(A == 34 && B == 55) Then what will be output.

   a) A == 34 && B == 55 b) A >= 30 || B <= 50 c) B= = 55 || A == 35

   d) A != 34 e) A >= 30 && A < 35 f) B>50 || B < 56

1. Mention the output of the following code at each step. ***in Memory***

```
int a,b,c ;
   a = 33;     b = 55;
   a += b;
   c = a;
   c -=b;
   c *= a;
    a++;
  System.out.print( "a= "+a + " ,b= " + b   + ", c= "+  c );
```

6. What is the result after the execution of the following program segment?

```
int a,b,c;
 a =  8;
 b = ++a  + 5;
 c = b--  + 10;
```

```
System.out.print( "a=  "+a + " ,  b= "+ b + ", c = "+ c );
```

7. Write down the output of following code and also justify your answer
   a) `System.out.print( 4 + 2 - 12 * 3 );` //what will be output
   b) `System.out.print( 4 + (2 – 12) * 3) ;` //what will be output
   c) `System.out.print( ( (4 + 2) - 12 ) * 3 )` //what will be output
8. What value is assigned to ***ReturnValue*** for each of the following expression and you decide ***ReturnValue*** Variable type.
   a) ReturnValue = 2 == 3;
   b) ReturnValue = "2" + "3";
   c) ReturnValue = 2 > 3;
   d) ReturnValue = 2 < 3;
   e) ReturnValue = (2 > 3) && (2 < 3) ;
   f) ReturnValue = (2 > 3) || (2 < 3)

Objective MCQ's

1. What is the value of the expression 4 * 2 + 3?
   a) 11
   b) -11
   c) 20
   d) 24

2. The logical And (&&) operator return TRUE if _____
   a) The left operand return a value is true
   b) The right operand return a value is true
   c) The left and right operand both return a value true is true
   d) The left and right operand both return a value true is false

3. Which arithmetic operator can be used as both prefix and postfix operators?
   a) ++
   b) –
   c) +
   d) *

4. Which of the following values can be assigned to a Boolean variable?
   a) True or False
   b) False
   c) 1
   d) Yes

5. What value is assigned to the ***ReturnValue*** variable in the statement
   ReturnValue = 100 **!=** 200 ;
   a) True
   b) False
   c) 100
   d) 200

6. Which of the following in an example of initializing a variable?
   a) num = 2;
   b) num >= 2;
   c) num <= 2;
   d) num == 2;

7. A relational operator is used to
   a) Combine the values
   b) Distinguish different types of variables.
   c) Change variables to logical values
   d) Compare the values

8. Operators are used to perform some operation on the _____.
   a) Operands
   b) Function
   c) Variables
   d) Constant

9. Operators are Concatenation or combine the different types of value and create single value
   a) X
   b) =
   c) +
   d) ++
10. Operators perform manipulations of data at bit level that is type of.
   a) Logical operators
   b) Relational Operators
   c) Conditional Operators
   d) Bitwise operators