

3. Fundamentals of Java Programming

Lesson 4: Operators

3.7. Operators

- An operator is a function that has a special symbolic name and is invoked by using that symbol with an expression.
- Java has several kinds of operators:
 - Arithmetic Operators
 - These are used to perform standard arithmetic operations performed on numbers
 - Assignment Operator
 - This is used to assigns a variable with a value
 - Logical (Boolean) Operators
 - These are used to perform standard logical operations on Boolean values
 - Bitwise Operators
 - These perform operations on the individual bits of integers
 - Conditional Operator
 - This acts as a shorthand for if-then-else

3.7. Operators

3.7.1. Arithmetic Operators

Symbol	Meaning	Example	
+	Addition	3+4	=7
-	Subtraction	15-7	=8
*	Multiplication	7*8	=56
/	Division	15/3	=5
%	Modulus	59%7	=3

3.7. Operators

3.7.2. Assignment Operator

- Assignment Operator (=)
- For example:

```
int a;  
a = 5;
```
- Right side of Assignment operator is evaluated first and then this value is assigned to the Left side of the Assignment operator

3.7. Operators

3.7.3. Logical (Boolean) Operators

Symbol	Meaning	Example	
&&	Logical AND	true && false	= false
	Logical OR	true false	= true
!	Logical NOT	!true	= false

3.7. Operators

3.7.4. Bitwise Operator

Symbol	Meaning	Example	
&	Bitwise AND	0 & 1	= 0
	Bitwise OR	0 1	= 1
^	Bitwise XOR	^0	= 1
<<	Left Shift	1<<4	= 16
>>	Right Shift	256>>4	= 64
~	Bitwise Compliment		

3.7. Operators

3.7.5. Assignment Shortcuts

- Often in programming, we have to apply some operator to some variable and then assign the resultant value back to the same variable. For example,
`a = a + 5;`
- Note, here we have two operations, one addition (`a + 5`) and one assignment (`a = ...`).
- Java provides shortcut operators that combine the two operations.
`a += 5;`
- Similarly we have, `-=`, `*=`, `/=`, `%=`, `&=`, `|=` etc.
- Very often we need to perform additions or subtractions by one (increments or decrements), that is expressions of the form `a = a + 1` or `a = a - 1`. These can be done by the Increment Operator and Decrement Operator `++` and `--`.
`a++;`
`a--;`

3.7. Operators

3.7.6. Relational Operator

Symbol	Meaning	Example	
==	Equal to	2==3	= false
!=	not Equal to	2!=3	= true
>	Greater than	2>3	= false
<	Less than	2<3	= true
>=	Greater than or Equal	2>=3	= false
<=	Less than or Equal	2<=3	= true

3.7. Operators

3.7.7. Conditional Operator

- Conditional Operator (?:)
- This operator has the form:
 - **(condition)?(value if true):(value if false)**
 - The condition is evaluated and if it is true **value if true** is returned, otherwise **value if false** is returned.
- For example, if **a** and **b** were integers, the following would return the maximum of **a** and **b**.
 - **(a>b)?a:b**
- This is a short hand for “if-then-else”

3.7. Operators

3.7.8. Operator Precedence

- Consider the following expression

$$6+2*3$$

- This consists of two operations, one addition and one multiplication.
- The value of the expression depends on which of these operations is performed first.
 - If addition is performed first we get:
 $6+2*3 = (6+2)*3 = 8*3 = 24$
 - If multiplication is performed first we get:
 $6+2*3 = 6+(2*3) = 6+6 = 12$
- To avoid ambiguity and confusion, java defines a clear order in which operators are evaluated. This is known as **operator precedence**.
- According to operator precedence, the multiplication operator (*) has higher precedence than the addition operator (+). Hence, the correct evaluation of $6+2*3$ is **12**.

3.7. Operators

3.7.8. Operator Precedence cont..

- The complete operator precedence is as follows.
- Operators at the top have higher precedence.
- Operators at the same level have equal precedence

Highest



Lowest

. [] ()	
++ -- ! ~	Increment Operators
new	
* / %	Arithmetic Operators
+ -	
<< >> >>>	Bitwise Shift Operators
< > <= >=	Comparison Operators
== !=	
&	Bitwise Operators
^	
&&	Logical Operators
?:	Conditional Operator
= += -= etc.	Assignment Operators
&= = <=< etc.	