**2.3 Java Operators:-**

Now that you've learned how to declare and initialize variables, you probably want to know how to *do something* with them. Learning the operators of the Java programming language is a good place to start. Operators are special symbols that perform specific operations on one, two, or three *operands*, and then return a result.

As we explore the operators of the Java programming language, it may be helpful for you to know ahead of time which operators have the highest precedence. The operators in the following table are listed according to precedence order. The closer to the top of the table an operator appears, the higher its precedence. Operators with higher precedence are evaluated before operators with relatively lower precedence. Operators on the same line have equal precedence. When operators of equal precedence appear in the same expression, a rule must govern which is evaluated first. All binary operators except for the assignment operators are evaluated from left to right; assignment operators are evaluated right to left.

|  |  |
| --- | --- |
| **Operator Precedence** | |
| **Operators** | **Precedence** |
| postfix | expr++ expr-- |
| unary | ++expr --expr +expr -expr ~ ! |
| multiplicative | \* / % |
| additive | + - |
| shift | << >> >>> |
| relational | < > <= >= instanceof |
| equality | == != |
| bitwise AND | & |
| bitwise exclusive OR | ^ |
| bitwise inclusive OR | | |
| logical AND | && |
| logical OR | || |
| ternary | ? : |
| assignment | = += -= \*= /= %= &= ^= |= <<= >>= >>>= |

In general-purpose programming, certain operators tend to appear more frequently than others; for example, the assignment operator "=" is far more common than the unsigned right shift operator ">>>". With that in mind, the following discussion focuses first on the operators that you're most likely to use on a regular basis, and ends focusing on those that are less common. Each discussion is accompanied by sample code that you can compile and run. Studying its output will help reinforce what you've just learned.

1. [**Simple Assignment Operator**](http://www.roseindia.net/java/master-java/simple-assignment-oprator.shtml)  
   Assignment operator is the most common operator almost used with all programming languages.
2. [**Arithmetic Operators**](http://www.roseindia.net/java/master-java/arithmetic-operators.shtml)   
   Arithmetic Operators are used to perform some mathematical operations like addition, subtraction, multiplication, division, and modulo (or remainder).
3. [**Unary Operators**](http://www.roseindia.net/java/master-java/unary-operators.shtml)  
   The unary operators requires only one operand to perform different kind of operations such as increasing/decreasing a value, negating an expression, or inverting a boolean value.
4. [**Equality and Relational Operators**](http://www.roseindia.net/java/master-java/Equality-relational-operators.shtml)   
   Whenever we need to compare the results of two expressions or operands in a program then the **equality** and **relational operators** are used to know whether an operand is equal, not equal, greater than, less than to another operand.
5. [**Conditional (Logical) Operators**](http://www.roseindia.net/java/master-java/conditional-operators.shtml)  
   Conditional operators return a **true** or a **false** value based on the state of the variables i.e. the operations using conditional operators are performed between the two boolean expressions.
6. [**Bitwise and Bit Shift Operators**](http://www.roseindia.net/java/master-java/bitwise-bitshift-operators.shtml)  
   In Java the bitwise and bit shift operators are used to manipulate the contents of variables at a bit level according to **binary** format.
7. [**Type Operators**](http://www.roseindia.net/java/master-java/TypeComparisonOperator.shtml)   
   Java provides a run-time operator **instanceof** to compare a **class** and an **instance** of that class. This operator " **instanceof**" compares an object to a specified class type
8. [**Operator Precedence**](http://www.roseindia.net/java/master-java/operator-precedence.shtml)   
   In Java,Operator Precedence is an **evaluation order** in which the operators within an expression are evaluated on the priority bases.