



OPERATORS

Category	Operator	Associativity
Postfix	<code>[] . dotoperator</code>	Left to right
Unary	<code>++ -- ! ~</code>	Right to left
Multiplicative	<code>* / %</code>	Left to right
Additive	<code>+ -</code>	Left to right
Shift	<code>>> >>> <<</code>	Left to right
Relational	<code>> >= < <=</code>	Left to right
Equality	<code>== !=</code>	Left to right
Bitwise AND	<code>&</code>	Left to right
Bitwise XOR	<code>^</code>	Left to right
Bitwise OR	<code> </code>	Left to right
Logical AND	<code>&&</code>	Left to right
Logical OR	<code> </code>	Left to right
Conditional	<code>?:</code>	Right to left
Assignment	<code>= += -= *= /= %= >>= <<= &= ^= =</code>	Right to left
Comma	<code>,</code>	Left to right

INSTANCE OF

```
public class Test {  
    public static void main(String args[]){  
        String name = "James";  
        // following will return true since name is type of String  
        boolean result = name instanceof String;  
        System.out.println( result );  
    }  
}
```

```
class Vehicle {}  
  
public class Car extends Vehicle {  
    public static void main(String args[]){  
        Vehicle a = new Car();  
        boolean result = a instanceof Car;  
        System.out.println( result );  
    }  
}
```



CONDITIONAL OPERATOR

```
variable x = (expression) ? value if true : value if false
```

```
public class Test {  
    public static void main(String args[]){  
        int a, b;  
        a = 10;  
        b = (a == 1) ? 20: 30;  
        System.out.println( "Value of b is : " + b );  
  
        b = (a == 10) ? 20: 30;  
        System.out.println( "Value of b is : " + b );  
    }  
}
```

