



JAVA

Review
Session

Java Operators

Operators are used to perform operations on variables and values.

The value is called an operand, while the operation (to be performed between the two operands) is defined by an operator:

Java divides the operators into the following groups:

Arithmetic operators

Assignment operators

Comparison operators

Logical operators

Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Examples:

$a+b$;

$a-b$;

a/b ;

$a\%b$;

$a*b$;

Java Assignment Operators

Assignment operators are used to assign values to variables.

Examples:

```
String a="test";
```

```
int b=23;
```

Java Comparison Operators

Comparison operators are used to compare two values.

Examples:

Equal to $x == y$

Not equal $x != y$

Greater than $x > y$

Less than $x < y$

Greater than or equal $x >= y$

Less than or equal $x <= y$

Logical operators

Logical operators are used to determine the logic between variables or values.

Examples:

Logical AND `&&` ex. `x < 5 && x > 10`

Logical OR `||` ex. `x < 5 || x > 4`

Logical NOT `!` ex. `!(x < 5 && x > 10)`

Switch Statement

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.

This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.

The following rules apply to a switch statement

1. The variable used in a switch statement can only be integers, convertible integers (byte, short, char), strings.
2. You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
3. The value for a case must be the same data type as the variable in the switch and it must be a constant or a literal.
4. When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
5. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.

The break Keyword

When Java reaches a break keyword, it breaks out of the switch block.

This will stop the execution of more code and case testing inside the block.

When a match is found, and the job is done, it's time for a break. There is no need for more testing.

A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

The default Keyword

The default keyword specifies some code to run if there is no case match:

Syntax for break keyword & default

```
switch(expression) {  
    case value :  
        // Statements  
        break;  
  
    case value :  
        // Statements  
        break;  
    // You can have any number of case statements.  
    default :  
        // Statements  
}
```