



Expressions, Statements Control Flow and Repeat

Sisoft Technologies Pvt Ltd
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad
Website: www.sisoft.in Email: info@sisoft.in
Phone: +91-9999-283-283

Expressions, Statements, and Blocks

The expressions : An expression may be formed by combining variables, operators and literals. It is created according to the language Syntax that evaluate the some value.

```
Example : int number=10;           //number=10 is expression.
```

The statements: When you write the complete expression with semicolon(;) it makes the statement.

The assignment expressions, Any use of ++ or --,

```
Example : number= 100; //assignment statement  
number++; //increment statement
```

The blocks: When your group zero or more statements by using the some balanced braces then it forms the block. So you may tell a block is a group of any number of statements.

Example :

```
if (some condition) { // begin block 1  
    System.out.println("Condition is true."); } // end block one  
else { // begin block 2  
    System.out.println("Condition is false.");  
} // end block 2
```

Control Flow Statements

The decision making-statements (if-then, if-then-else & switch), looping statements (while, do-while & for) and the branching statements (break, continue and return) are control flow statements.

The control statements

- Selection Statements :
 - also known decision making-statements (if-then, if-then-else & switch)
- Repetition Statements :
 - also known as looping statements (while, do-while & for)
- Branching Statements:
 - also known as break, continue and return

Control Flow Statements Continue..

Selection Statements :

If Statement:

- This is a control statement to run a block of code or a single statement, when the given condition is TRUE, and if it is FALSE then it skips if the block and rest code of the program is executed.
- The syntax of IF-Statement:

```
if(Conditional_Expression) {  
    <Statements>;  
...;}
```

The If-else statement:

- The 'if-else' statement is an extension of if statement that provides the another option when 'if'-statement seems to 'false' i.e. else block is executed if 'if'-statement is FALSE.

```
if(Conditional_Expression) {  
    <Statements>;} else{  
    <Statements>;}
```

Control Flow Statements Continue..

Selection Statements :

Switch Statement:

- This is simpler implementation of the if-else statements. The keyword 'switch' is followed by an expression that should seem to be byte, char, short or int primitive data-types.
- Syntax:

```
switch(Control_Expression) {  
    case expression 1:  
        <Statement>;  
    .....  
    case expression n:  
        <Statement>;  
    default:  
        <Statement>;  
} //end switch
```

Control Flow Statements Continue..

Repetition Statements :

while-loop statements:

- This is a repeating statement or looping. It runs statements or a block of code till the given condition is TRUE. The expression must be evaluated to a boolean value.
- It continues testing the condition and runs the block of code. When the expression results to FALSE control comes out of the loop.

- **Syntax:**

```
while(Expression) {  
    <Statement>;  
    ...;  
}
```

- **Example:**

```
int i = 1; //This will print the 1 to 10  
while (i <= 10){  
    System.out.println("Number = " + i);  
    i++; }  

```

Control Flow Statements Continue..

Repetition Statements :

Do-while-loop statements:

- This is another looping statement that tests the given condition past so you may tell that the do-while looping statement is the past test loop statement.

- Syntax:

```
do {  
    <Statement>;  
    .....;  
}while (Expression);
```

Example:

```
int i = 1;  
do{  
    System.out.println("Number = " + i);  
    i++;  
}while(i <= 10);
```

Control Flow Statements Continue..

Repetition Statements :

For-loop statements:

- This is the loop statement that provides the compact way to iterate over the range of values.

- **Syntax:**

```
for (Initialization; Condition; Increment or Decrement){  
    <Statement>;  
    .....;  
}
```

- **Initialization:** The loop is begun with the value specified.
- **Condition:** It evaluates to either 'TRUE' or 'FALSE'. Whether it is FALSE then the loop is terminated.
- **Increment or Decrement:** After the every iteration, value increments or decrements.

Example:

```
for (int number = 1; number <= 10; number++){  
    System.out.println("Number --> " + number );}
```


Control Flow Statements Continue..

Branching Statements :

Break statements:

- The break statement is the branching statement that contains the two forms - Labeled and Unlabeled.
- The break statement is used for breaking the running state of a loop (while, do-while & for).
- It finish the switch statements.
- **Syntax :**
 - `break;` // The breaks switch statement or innermost loop.
 - `break label;` // It breaks the outermost loop into the series of nested loops.
- When if statement evaluates to TRUE it prints 'The data is found' and comes out of the loop and runs the statements just following the loop.

Control Flow Statements Continue..

Branching Statements :

Continue statements:

- This is the branching statement that are used in the looping statements (while, do-while & for) to neglect the current iteration of the loop and resume next iteration.

The syntax:

```
continue;
```

The return statements: It's a special branching statement that shifts the control to the caller of a method. This statement is used to return the value to the caller method and finish the execution of a method.

Syntax:

```
return;
```

```
return values;
```