

Assignment - 01

CSA0992-

Programming in
JAVA for Freshers.

Name:

K.V. Sai Sanjana

Reg. No: 192011124

Dept: CSE

Operators :-

- * Java provides many types of operators which can be used according to the need.
- * They are classified based on the functionality they provide.
- * Operators in Java are the symbols used for performing specific operators in Java.
- * Operators makes tasks like addition, multiplication, etc. which look easy although the implementation of these tasks is quite complex.

Types of Operators:-

1. Arithmetic operators
2. Assignment operators
3. Comparison Operators
4. Logical operators
5. Bitwise operators
6. Increment / Decrement operators.

Arithmetic Operators:-

They are used to perform simple arithmetic operations on primitive data types.

* : Multiplication

/ : Division

% : Modulo

+ : Addition

- : Subtraction.

Operator	Name	Example
*	Multiplication	$x * y$
+	Addition	$x + y$
-	Subtraction	$x - y$
/	Division	x / y
%	Modulos	$x \% y$

Assignment Operator:-

Operator	Example
=	$x = 5$
+=	$x += 3$
-=	$x -= 3$
*=	$x *= 5$
/=	$x /= 5$
%=	$x \% = 5$
f=	$x f= 5$
!=	$x != 3$
^=	$x ^= 3$
>>=	$x >>= 5$
<<=	$x <<= 5$

Comparison Operators:-

Operator	Name	Example
$= =$	Equal to	$x = = 5$
\neq	Not equal to	$x \neq 5$
$>$	Greater than	$x > y$
$<$	Less than	$x < y$
\geq	Greater than or equal to	$x \geq 5$
\leq	Less than or equal to	$x \leq 5$

Logical Operators:-

Operators	Name	Example
$\&\&$	Logical and	$x < 5 \&\& x < 10$
$\ $	Logical or	$x < 5 \ x < 4$
!	Logical Not	$!(x < 5 \&\& x < 10)$

Bitwise Operators:-

Operator	Name	Example
$\&$	Bitwise AND	$x \& y$
\wedge	Bitwise Exclusive OR	$x \wedge y$

<code>~</code>	Bitwise inclusive OR	$x \mid y$
<code>~</code>	Compliment	$\sim x$
<code><<</code>	Left shift	$x << y$
<code>>></code>	Right shift	$x >> y$

Increment / Decrement Operations :-

Operator	Name	Example
<code>++</code>	Post Increment Pre Increment	$x++$ $++x$
<code>--</code>	Post Decrement Pre Decrement	$x--$ $--x$

Control Statements:-

- * Executed according to order.
- * Smooth flow of program.

Types:-

Decision Making Statements:-

- * If statements
- * Switch statements.

Looping Statements:-

- * Do while
- * While
- * For loop.

Jump Statements:-

- * Break statement
- * Continue statement.

Decision Making Statements

If statements:-

Evaluate a condition
Derived specific condition
Condition enter either True or False

Types:-

- * Simple If statement
- * If-else statement.
- * If-else-If ladder
- * Nested If statement.

Simple If statement:-

Expression evaluates to true

Syntax:

```
if (condition)
{
    Statement;
}
```

If - Else Statement :-

If (condition)

```
{ Statement 1 ;  
}
```

else

```
{ Statement 2 ;  
}
```

Nested If Statement :-

If (condition 1)

```
Statement 1 ;  
}
```

If (condition 2)

```
Statement 2 ;  
}
```

else

```
Statement 3 ;  
}
```

 { Statement 3 ;
 }
}

Switch Statements :-

Multiple blocks of code in single code

Switch (expression)

```
{  
    case value 1 ;  
        Statement 1 ;  
        break ;  
    }  
    case value 2 ;  
        Statement 2 ;  
        break ;  
    default ;  
        default statement ;  
}
```

If - else - If ladder

If (condition)

```
{ Statement 1 ;  
}
```

else if (condition 2)

```
{ Statement 2 ;  
}
```

else

```
{ Statement 3 ;  
}
```

Looping statements:-

execute code repeatedly

Execution instruction particular condition

Types :-

* for loop

* while loop

* do while loop.

For Loop:-

```
for (initialization; condition; increment/decrement)
{
    block of statements;
}
```

While Loop:-

```
while (condition)
```

```
{
    statement;
```

```
}
```

Do - while :-

```
do
```

```
{
```

```
    statement
```

```
}
```

```
while (condition);
```

Jump Statement:-

Transfer control specific statement

Execute other parts of the program.

Types:-

- * Break :- Stop the current flow of the program.
- * Continue :- Skips the specific part.

Applications:-

- * Mathematical calculation.
- * Searching
- * Sorting
- * Handling Electronic and IoT related Operations.