



OPERATOR AND LOOPS



Assignment Question

1. What are the Conditional Operators in Java?

Answer → Conditional Operators in Java are used to work on conditions. It is a unique operator used in place of 'If-else' statements.

There are three types of the conditional operators in Java:

1. Conditional AND
2. Conditional OR
3. Conditional NOT

Conditional AND :-

The output is only true whenever both the expressions surrounding the operator are true.

Expr 1	Expr 2	Expr 1 && Expr 2
T	T	T
T	F	F
F	T	F
F	F	F

Conditional OR :-

The output is only true whenever either of the expressions surrounding the operator is true.

Expr 1	Expr 2	Expr 1 Expr 2
T	T	T
T	F	T
F	T	T
F	F	F



Conditional NOT :-

The output is reverse whenever there is true is convert into false or vice versa.

Expr 1	Expr 1 !
T	F
F	T

2. What are the types of operators based on the number of operands?

Answer → Operators can be classified based on the number of operands they require into unary (one operand), binary (two operands), and ternary (three operands) operators.

Unary Operators :-

These operators operate on a single operand.

- **Examples:-** Increment (++), Decrement (--), Negation (-), Logical NOT (!) .

Binary Operators :-

These operators operate on two operands.

- **Examples:** Arithmetic operators (+, -, *, /), Comparison operators (==, !=, >, <), Logical operators (&&, ||), Assignment operators (=).

Ternary Operators :-

These operators, also known as conditional operators, operate on three operands.

- **Example:** The conditional (ternary) operator (? :).

3. What is the use of Switch case in Java programming?

Answer → The switch case in java is used to select one of many code blocks for execution.

Break keyword :- As java reaches a break keyword, the control breaks out of the switch block. The execution of code stops on encountering this keyword, and the case testing inside the block ends as the match is found.

Java

```
switch (expression) {  
    case value1:  
        // Code to be executed if expression matches value1  
        break; // optional, but often used to prevent fall-through  
    case value2:  
        // Code to be executed if expression matches value2  
        break;  
    // ... more cases  
    default:  
        // Code to be executed if no case matches  
        break;  
}
```

- Switch Case is used When you need to execute different code blocks based on the value of a variable or expression.

4. What are the priority levels of arithmetic operation in Java?

Answer → In Java, arithmetic operations follow a standard precedence order: multiplication, division, and modulus operators have higher precedence than addition and subtraction operators .

Division , Multiplication and Modulus come on 5 Level in Precedence Tables.

While Addition and Subtraction come on 6 Level in Precedence Tables.

Clearly illustrate in the below table with Associativity Order in Table.



<code>[] . ,</code>	subscript, member selection, comma	left-associative
<code>x++ x-- ~x</code>	postfix increment, postfix decrement, bitwise negation	
<code>++x --x +x -x !x</code>	prefix increment, prefix decrement, unary positive, unary negative, logical negation	right-associative
<code>(X) new X</code>	typecasting, object creation	
<code>* / %</code>	multiplication, division, modulus	left-associative
<code>x+y x-y x+"x"</code>	addition, subtraction, string concatenation	
<code><< >> >>></code>	bitwise shift	
<code>< <= > >=</code>	comparison	
<code>instanceof</code>	runtime type compatibility	
<code>== !=</code>	equality and inequality	
<code>&</code>	bitwise AND	
<code>^</code>	bitwise XOR	
<code> </code>	bitwise OR	
<code>&&</code>	logical AND	
<code> </code>	logical OR	
<code>x ? y : z</code>	ternary (conditional)	right-associative
<code>+= -= *= /= %= <<= >>= >>>= &= ^= =</code>	assignment and compound assignment	



5. What are the conditional Statements and use of conditional statements in Java?

Answer → In Java, conditional statements, also known as decision statements, allow you to control the flow of your program by executing different code blocks based on whether a condition is true or false, using keywords like if, else, else if, and switch.

1. If :-

if statement Executes a block of code only if a specified condition is true.

```
if (condition) {  
    // Code to execute if the condition is true  
}
```

2. If – Else :-

If-else statement: Executes one block of code if the condition is true and another block if the condition is false.

```
if (condition) {  
    // Code to execute if the condition is true  
} else {  
    // Code to execute if the condition is false  
}
```

3. If – Else If – Else :-

if-else if-else statement (else-if ladder): Allows you to check multiple conditions sequentially, executing the block associated with the first true condition encountered.

```
if (condition1) {  
    // Code to execute if condition1 is true  
} else if (condition2) {  
    // Code to execute if condition2 is true  
} else if (condition3) {  
    // Code to execute if condition3 is true  
} else {  
    // Code to execute if none of the conditions are true  
}
```

6. What is the syntax of if else statement?

Answer → Syntax: The syntax of the if-else statement consists of the if keyword followed by a condition in parentheses, followed by a block of code to be executed if the condition is true. Optionally, an else block can follow, containing code to be executed if the condition is false.

Example :-

```
If(Condition)
{
    // Block Of Code
}
Else
{
    //Block Of Code
}
```

Note :- If Block Of Code Executed when Condition of If Become True Otherwise Else Block Code is Executed.

7. What are the 3 types of iterative statements in java ?

Answer → The three types of iterative (or looping) statements in Java are the for loop, the while loop, and the do-while loop.

1. For Loop :- Used when you know the number of iterations in advance or have a clear initialization, condition, and update expression.

Syntax:- for(initialization; condition; update)

```
{
    // code to be executed
}
```

2. While Loop :- Used when you want to repeat a block of code as long as a condition is true.

Syntax:- while (condition)

```
{
    // code to be executed
    Updation;
}
```

3. Do – While Loop :- Similar to the while loop, but guarantees that the code block inside the loop will execute at least once before the condition is checked.

Syntax:- do
 {
 // code to be executed
 } while (condition);

8. Write the difference between for loop and do-while loop ?

Answer →

S.No.	For Loop	Do – While Loop
1.	Syntax:- For(initialization; condition; updating), { // Statements; }	Syntax: Do { //Statements; } While(condition);
2.	It is known as entry controlled loop.	It is known as exit controlled loop.
3.	If the condition is not true first time than control will never enter in a loop.	Even if the condition is not true for the first time the control will enter in a loop.
4.	There is no semicolon; after the condition in the syntax of the for loop.	There is semicolon; after the condition in the syntax of the do while loop.
5.	Initialization and updating is the part of the syntax.	Initialization and updating is not the part of the syntax.
6.	For loop is use when we know the number of iterations means where the loop will terminate.	Do while loop is use when we don't know the number of iterations means where the loop will terminate.

9. Write a program to print numbers from 1 to 10?

Answer →

```
public class Number
{
    public static void main(String[] args)
    {
        for(int i=1;i<=10;i++)
        {
            System.out.println(i);
        }
    }
}
```