

## 1.What are the conditional operators in java ?

### ANSWER:-

They were used when a condition comprises more than one boolean expression.

We have three types of conditional operators

a)logical-and b)logical-or c)ternary operator

### Logical -and operator(&&):-

It is used when we want the condition to be true iff both the expressions are true.

### SYNTAX:-

```
if(condition - 1 && condition - 2) {  
    statement;  
}
```

### Example code:-

```
class Test{  
    public static void main(String []args){  
        int n=7;  
        if(n>5 && n<10){  
            System.out.println("true");  
        }  
        else{  
            System.out.println("false");  
        }  
    }  
}
```

OutPut:-

true

### Logical -or operator(||):-

This operator is used when we are satisfied as long as one of the boolean value is evaluated as true.

### SYNTAX:-

```
if(condition - 1 || condition - 2) {  
    statement;  
}
```

### Example code:-

```
class Test{  
    public static void main(String []args){  
        int n=7;  
        if (val < 5 || val > 10) {  
            System.out.print(val);  
        }  
        else{  
            System.out.println("not in range");  
        }  
    }  
}
```

OutPut:-

7

### **Ternary Operator(?:):-**

It is a smaller version for the if-else statement. If the condition is true then the statement - 1 is executed else the statement - 2 is executed.

### **SYNTAX:-**

condition ? statement - 1 : statement - 2;

### **Example Code:-**

```
public class TernaryOpe {
    public static void main(String []args){
        int a=10;
        int b=20;
        int c=30;
        int res=(a<b)?a:b;
        System.out.println(res);
        String res1=(a>b)? "it is less" : "it is greater";
        System.out.println(res1);
        int res3= (a<b)?((a<c)?a:c):(b<c)?b:c;
        System.out.println(res3);
    }
}
```

### **Output:-**

```
10
it is greater
10
```

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

### **ANSWER:-**

1.Unary Operators: Operators that operate on a single operand. Examples include the unary minus operator (-), increment operator (++), and decrement operator (--).

2.Binary Operators: Operators that operate on two operands. Examples include the addition operator (+), subtraction operator (-), and multiplication operator (\*).

3.Ternary Operators: Operators that operate on three operands. The ternary operator (? :) is the only ternary operator in Java.

## **3.What is the use of the switch case in java programming ?**

### **ANSWER:-**

The switch case statement in Java is used as a control structure to execute a certain block of code based on the value of a variable. It is a multi-way decision-making statement that provides a convenient and efficient alternative to writing multiple if-else statements.

The switch case statement works by matching the value of the variable with a specified case constant. If a match is found, the code inside that case block is executed. If no match is found, the default block is executed.

The switch case statement is particularly useful in situations where there are multiple conditions to be checked and multiple actions to be performed based on those conditions. It provides a clean and readable way to handle complex control flow in Java programs.

CODE:-

```
int day = 2;
```

```
switch (day) {  
    case 1:  
        System.out.println("Monday");  
        break;  
    case 2:  
        System.out.println("Tuesday");  
        break;  
    case 3:  
        System.out.println("Wednesday");  
        break;  
    default:  
        System.out.println("Invalid day");  
}
```

Output is Tuesday.

#### **4.What are the priority levels of arithmetic operations in java ?**

**ANSWER:-**

The priority levels of arithmetic operations in Java are as follows:

- 1.Parentheses ( )
- 2.Exponentiation (^)
- 3.Unary operations (such as unary minus or positive)
- 4.Multiplication and Division (\*, /)
- 5.Addition and Subtraction (+, -)

When multiple operations are present in an expression, Java performs operations with higher priority first and then moves to the lower priority operations. If the operations have the same priority, they are performed from left to right.

#### **5.What are the conditional statements and use of conditional statements in java ?**

**ANSWER:-**

Conditional statements in Java are control structures that allow the program to make decisions based on certain conditions. The two main types of conditional statements are the if statement and the switch statement.

##### **1.The if statement:**

The if statement is used to execute a block of code only if a specified condition is true. It has the following syntax:

```
if (condition) {  
    // code to be executed if condition is true  
}
```

```
if (condition) {  
    // code to be executed if condition is true  
} else {  
    // code to be executed if condition is false  
}
```

The switch statement:

The switch statement is used to execute a block of code based on the value of an expression. It has the following syntax

```
switch (expression) {  
    case value1:  
        // code to be executed if expression equals value1  
        break;  
    case value2:  
        // code to be executed if expression equals value2  
        break;  
    ...  
    default:  
        // code to be executed if no case matches  
}
```

**6.What is the syntax of if else statement ?**

**ANSWER:-**

```
if(condition){  
    //Statements  
}  
else{  
    //Statements  
}
```

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

**ANSWER:-**

The three types are

- a)for loop
- b)while loop
- c)do-while loop

**a)while loop:-**

A while loop is a loop that runs through its body, known as a while statement, as long as a predetermined condition is evaluated as true.

**Syntax:-**

```
while (condition)
statement;
```

**Code:-**

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

**Output:-1 2 3 4 5 6 7 8 9 10**

**b)for loop:-**

Unlike while loop, in for loop we have 3 parts in the for header .  
init-statement    condition    final-expression

**Syntax:-**

```
for (init-statement; condition; final-expression) {
statement
}
```

**Code:-**

```
for (int index = 0; index != 5; index++) {
System.out.print(index + " ");
}
```

**Output:-0 1 2 3 4**

**c)do-while loop:-**

Unlike while and for loop, do-while loop tests for the condition at the end of each execution for the next

iteration. In other words, the loop is executed at least once before the condition is checked.

Other than that

everything is the same as in the while loop.

**Syntax:-**

```
do {
statement;
} while (condition);
```

**Code:-**

```
int idx = 15;
do {
System.out.print(idx + " ");
} while (idx < 5);
```

**output:-15**

## 8.What are the differences between while and do-while loop?

### ANSWER:-

The differences between while and do-while loop are:

#### 1.Execution:

The while loop executes the code block only if the condition is true, while the do-while loop executes the code block at least once, and then repeatedly if the condition is true.

#### 2.Condition check:

The while loop checks the condition before executing the code block, while the do-while loop checks the condition after executing the code block.

#### 3.Syntax:

The syntax for the while loop is:

```
while (condition) {  
    // code to be executed  
}
```

The syntax for do-while loop is

```
do {  
    // code to be executed  
} while (condition);
```

#### 4.Use cases:

The while loop is used when you want to repeat the code block if a condition is true, and you want to stop the execution if the condition becomes false. The do-while loop is used when you want to repeat the code block at least once, regardless of the condition, and then repeat it if the condition is true.

In conclusion, while and do-while loops serve similar purposes, but they differ in their execution and the way they check the condition. Both can be used to repeat a code block multiple times, and the choice between the two depends on the specific requirements of the program.

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

### ANSWER:-

Code:-

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

Output:-1 2 3 4 5 6 7 8 9 10