

Q1. What are the Conditional Operators in Java?

→ In Java, conditional operators check the condition and decides the desired result on the basis of both conditions.

There are three types of the conditional operator in Java:

- *Conditional AND
- *Conditional OR
- *Ternary Operator

<u>Operator</u>	<u>Symbol</u>
Conditional or Logical AND	&&
Conditional or Logical OR	
Ternary Operator	?:

1.Conditional AND:

The operator is applied between two Boolean expressions. It is denoted by the two AND operators (&&). It returns true if and only if both expressions are true, else returns false.

2.Conditional OR:

The operator is applied between two Boolean expressions. It is denoted by the two OR operator (||). It returns true if any of the expression is true, else returns false.

3.Ternary Operator:

The meaning of ternary is composed of three parts. The ternary operator (? :) consists of three operands. It is used to evaluate Boolean expressions. The operator decides which value will be assigned to the variable.

Q2. What are the types of operators based on the number of operands?

→ Types of operators are as follows:

- *Arithmetic Operators.
- *Assignment Operators.
- *Relational Operators.
- *Logical Operators.
- *Unary Operators.
- *Bitwise Operators.
- *Ternary Operator.

1.Arithmetic Operators in Java:

Arithmetic Operators are used to perform mathematical operations like addition, subtraction, etc.

<u>Operator</u>	<u>Description</u>	<u>Example</u>
+ Addition	Adds values on either side of the operator	A+B=30
- Subtraction	Subtracts the right-hand operator with left-hand operator	A-B=-10

* Multiplication	Multiplies values on either side of the operator	A*B=200
/ Division	Divides left hand operand with right hand operator	A/B=0
% Modulus	Divides left hand operand by right hand operand and returns remainder	A%B=0

2.Assignment Operators in Java:

An Assignment Operator is an operator used to assign a new value to a variable.

Operators	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b
+=	It adds right operand to the left operand and assigns the result to left operand	c += a
-=	It subtracts right operand from the left operand and assigns the result to left operand	c -= a
*=	It multiplies right operand with the left operand and assigns the result to left operand	c *= a
/=	It divides left operand with the right operand and assigns the result to left operand	c /= a
%=	It takes modulus using two operands and assigns the result to left operand	c %= a
^=	Performs exponential (power) calculation on operators and assign value to the left operand	c ^= a

3.Relational Operators in Java:

These operators compare the values on either side of them and decide the relation among them.

Operators	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(A == B) is not true

!=	If the values of two operands are not equal, then condition becomes true.	(A != B) is true
>	If the value of the left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true
<	If the value of the left operand is less than the value of right operand, then condition becomes true.	(a < b) is true
>=	If the value of the left operand is greater than or equal to the value of the right operand, then condition becomes true.	(a >= b) is not true
<=	If the value of the left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true

4.Logical Operators in Java:

The following are the Logical operators present in Java:

Operators	Description	Example
&& (and)	True if both the operands is true	a<10 && a<20
(or)	True if either of the operands is true	a<10 a<20
! (not)	True if an operand is false (complements the operand)	!(x<10 && a<20)

5.Unary Operator in Java:

Unary operators are the one that needs a single operand and are used to increment a value, decrement or negate a value.

Operators	Description	Example
++	increments the value by 1. There is post-increment and pre-increment operators	a++ and ++a
—	decrements the value by 1. There is post decrement and pre decrement operators	a– or –a

!	invert a boolean value	!a
---	------------------------	----

6.Bitwise Operator in Java:

Bitwise operations directly manipulate bits. In all computers, numbers are represented with bits, a series of zeros and ones.

Operators	Description	Example
& (AND)	returns bit by bit AND of input	a&b
(OR)	returns OR of input values	a b
^ (XOR)	returns XOR of input values	a^b
~ (Complement)	returns the one's complement. (all bits reversed)	~a

7.Ternary Operators in Java:

The ternary operator is a conditional operator that decreases the length of code while performing comparisons and conditionals. This method is an alternative for using if-else and nested if-else statements. The order of execution for this operator is from left to right.

Syntax:

(Condition) ? (Statement1) : (Statement2);
--

Condition: It is the expression to be evaluated which returns a boolean value.

Statement 1: It is the statement to be executed if the condition results in a true state.

Statement 2: It is the statement to be executed if the condition results in a false state.

Q3.What is the use of Switch case in Java programming?

→ Use of Switch Case in Java:

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.

Q4.What are the conditional Statements and use of conditional statements in Java?

→ Conditional statements in Java are the executable block of code (or branch to a specific code) dependent on certain conditions. These statements are also known as decision statements or selection statements in Java.

You can use these conditions to perform different actions for different decisions.

Java has the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to specify many alternative blocks of code to be executed

Q5.What is the syntax of if else statement?

→ **Syntax:**

```

If(condition){

    // code if condition is true

}else{

    //code if condition is false

}

```

Q6.How do you compare two strings in Java?

→ To compare two strings in java we use following approach:

1. == operator

It compare the reference of the object.

2. equals()

It compares the contents of two objects.

Example:

```

String s1 = new String ("sachin");
String s2 = new String ("sachin");
System.out.println(s1 == s2); // false
System.out.println(s1.equals(s2)); // true

```

Q7.What is Mutable String in Java Explain with an example?

→ Mutable strings are those strings whose content can be changed without creating a new object.

StringBuffer and StringBuilder are mutable versions of String in java.

Example:

```

class MutableStringDemo{
public static void main(String args[]){
StringBuffer sBuffer1=new StringBuffer("Welcome");
System.out.println("Original String is :: " + sBuffer1 + ":: having length " + sBuffer1.length());
//using append method
sBuffer1.append(" To Edubca");
System.out.println("Modified String after append is :: " + sBuffer1 + " :: having length " +
sBuffer1.length());
//using reverse method
sBuffer1.reverse();
System.out.println("Modified String after Reverse is :: " + sBuffer1);
}
}

```

```
}  
}
```

Q8. Write a program to sort a String Alphabetically

```
→  
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args)  
    {  
        String str;  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the string : ");  
        str = sc.nextLine();  
        int j = 0;  
        char temp = 0;  
        char[] chars = str.toCharArray();  
        for (int i = 0; i < chars.length; i++) {  
            for (j = 0; j < chars.length; j++) {  
                if (chars[j] > chars[i]) {  
                    temp = chars[i];  
                    chars[i] = chars[j];  
                    chars[j] = temp;  
                }  
            }  
        }  
        System.out.println("The sorted string is : ");  
        for (int i = 0; i < chars.length; i++) {  
            System.out.print(chars[i]);  
        }  
    }  
}
```

Output:

Enter the string :

hfgdbeca

The sorted string is :

Abcdefgh

Q9. Write a program to check if the letter 'e' is present in the word 'Umbrella'

```
→ public class check{  
    public static void main(String[] args) {  
        String a = "Umbrella";  
        boolean per = false;  
        for(int i = 0; i < a.length(); i++){  
            if(a.charAt(i) == 'e'){
```

```
        per = true;
        break;
    }
}
System.out.println(per);
}
```

Output:

true

Q10.Where exactly is the string constant pool located in the memory?

→ String constant pool is a pool of strings stored in a separate place within the Heap area.