

Q1. What are the Conditional Operators in Java?

In Java, the conditional operators are used to perform logical operations and make decisions based on conditions. The conditional operators are:

Logical AND (&&): It returns true if both operands are true, otherwise false.

Logical OR (||): It returns true if at least one of the operands is true, otherwise false.

Logical NOT (!): It is a unary operator that reverses the logical state of its operand. If the operand is true, it returns false, and if the operand is false, it returns true.

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

Based on the number of operands, operators in Java are categorized into three types:

Unary Operators: These operators work with a single operand. Examples include ++ (increment), -- (decrement), ! (logical NOT), etc.

Binary Operators: These operators work with two operands. Examples include + (addition), - (subtraction), \* (multiplication), / (division), % (modulus), etc.

Ternary Operator: There is only one ternary operator in Java, which is the conditional operator (? :). It works with three operands and is used for making decisions in expressions.

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

The switch case statement in Java is used to make multiple decisions based on the value of an expression. It provides a concise and efficient way to handle multiple cases without using multiple if-else statements. The switch case is used when there are multiple cases, and each case requires different code execution based on the value of the expression.

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

Conditional statements in Java are used to make decisions in a program based on certain conditions. They allow the program to execute different blocks of code based on the evaluation of a given condition. The conditional statements in Java are:

if statement: It executes a block of code if the given condition is true.

if-else statement: It executes one block of code if the given condition is true, and another block of code if the condition is false.

if-else-if ladder: It is used when there are multiple conditions to be checked in sequence, and only one of them needs to be executed.

switch case statement: It is used for making multiple decisions based on the value of an expression.

Q5. What is the syntax of if else statement?

The syntax of the if-else statement in Java is as follows:

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

Q6. How do you compare two strings in Java?

In Java, you can compare two strings using the equals() method or the compareTo() method. The equals() method checks if the contents of the two strings are equal, whereas the compareTo() method compares the lexicographical order of the strings.

Example using equals() method:

```
String str1 = "Hello";
String str2 = "World";

if (str1.equals(str2)) {
    System.out.println("Strings are equal.");
} else {
    System.out.println("Strings are not equal.");
}
```

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

In Java, a mutable string refers to a string that can be changed or modified after its creation. The StringBuilder and StringBuffer classes in Java are used to create mutable strings. Unlike the String class, which is immutable, the StringBuilder and StringBuffer classes allow you to modify the content of the string without creating a new object.

Example using StringBuilder:

```
StringBuilder sb = new StringBuilder("Hello");
sb.append(" World"); // Appends " World" to the existing string
System.out.println(sb.toString()); // Output: "Hello World"
```

Q8. Write a program to sort a String Alphabetically

Here's a Java program to sort a string alphabetically:

```
import java.util.Arrays;

public class StringSort {
    public static void main(String[] args) {
        String str = "programming";
        char[] chars = str.toCharArray();
        Arrays.sort(chars);
        String sortedString = new String(chars);
        System.out.println("Original String: " + str);
        System.out.println("Sorted String: " + sortedString);
    }
}
```

Original String--- programming  
Sorted String----- agginmnoorr

