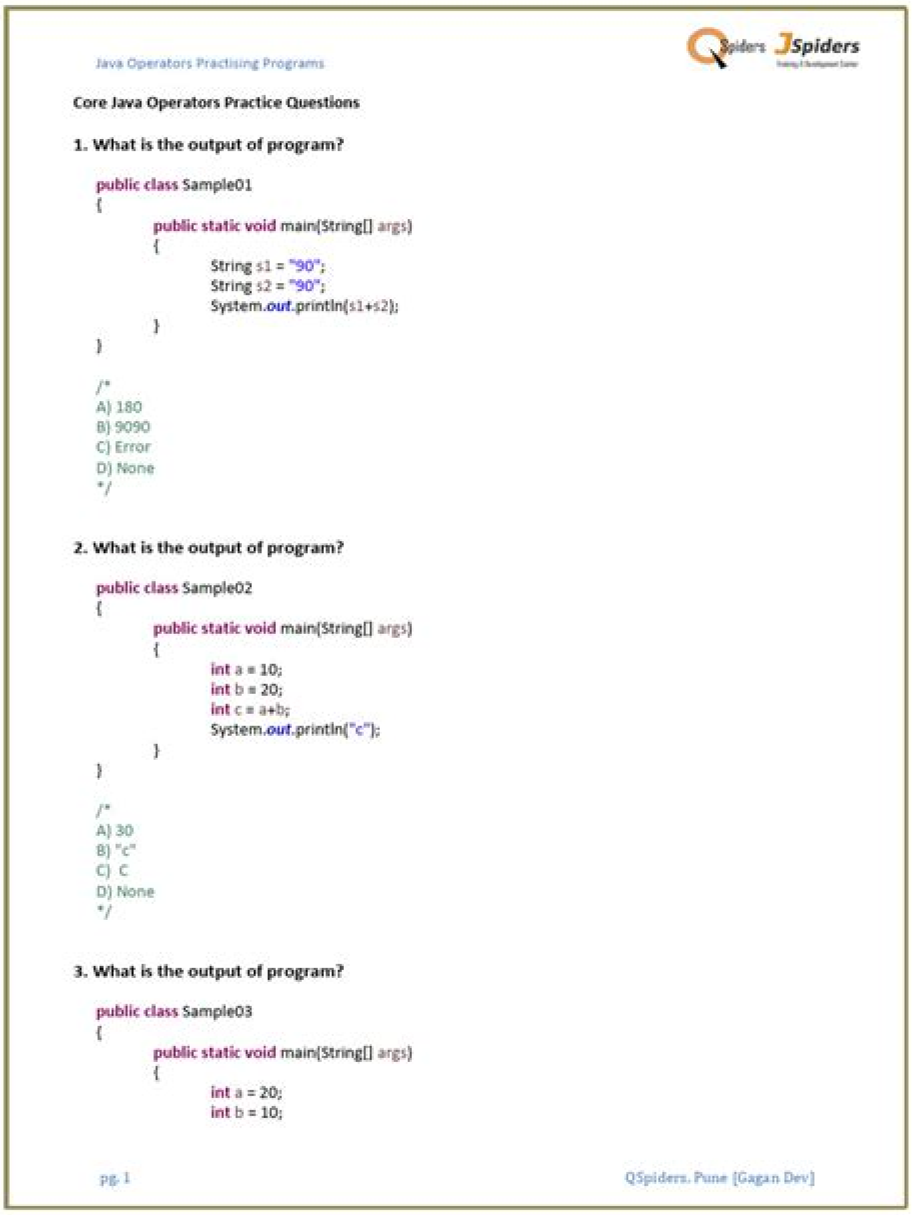
Java Operators Practice Questions



Java Operators Practising Programs

Core Java Operators Practice Questions

1. What is the output of program?

```
public class Sample01
{
        public static void main(String[] args)
        {
                String s1 = "90";
                String s2 = "90";
                System.out.println(s1+s2);
        }
}

/*
A) 180
B) 9090
C) Error
D) None
*/
```

2. What is the output of program?

```
public class Sample02
{
        public static void main(String[] args)
        {
                int a = 10;
                int b = 20;
                int c = a+b;
                System.out.println("c");
        }
}

/*
A) 30
B) "c"
C)  c
D) None
*/
```

3. What is the output of program?

```
public class Sample03
{
        public static void main(String[] args)
        {
                int a = 20;
                int b = 10;
```

pg. 1 QSpiders. Pune [Gagan Dev]

Java operators practice questions are essential for anyone looking to master Java programming. Operators in Java are special symbols that perform operations on variables and values. Understanding how to use these operators effectively can improve your coding skills and help you write efficient Java applications. In this article, we will explore various types of operators in Java, followed by a collection of practice questions that will aid your learning.

Types of Operators in Java

Java supports several types of operators, each serving a unique purpose. Below are the primary categories:

1. Arithmetic Operators

These operators are used for performing basic mathematical operations.

- Addition (+): Adds two operands.
- Subtraction (-): Subtracts the second operand from the first.
- Multiplication (): Multiplies two operands.
- Division (/): Divides the numerator by the denominator.
- Modulus (%): Returns the remainder of a division operation.

2. Relational Operators

Relational operators are used to compare two values.

- Equal to (==): Checks if two operands are equal.
- Not equal to (!=): Checks if two operands are not equal.
- Greater than (>): Checks if the left operand is greater than the right.
- Less than (<): Checks if the left operand is less than the right.
- Greater than or equal to (>=): Checks if the left operand is greater than or equal to the right.
- Less than or equal to (<=): Checks if the left operand is less than or equal to the right.

3. Logical Operators

Logical operators are used to combine multiple boolean expressions.

- Logical AND (&&): Returns true if both operands are true.
- Logical OR (||): Returns true if at least one operand is true.
- Logical NOT (!): Reverses the logical state of its operand.

4. Bitwise Operators

These operators perform operations on bits and are primarily used in low-level programming.

- Bitwise AND (&): Performs a bitwise AND operation.
- Bitwise OR (|): Performs a bitwise OR operation.
- Bitwise XOR (^): Returns the bitwise exclusive OR of two operands.
- Bitwise Complement (~): Reverses the bits of its operand.
- Left Shift (<>): Shifts bits to the right, preserving the sign bit.

5. Assignment Operators

Assignment operators are used to assign values to variables.

- Simple Assignment (=): Assigns the value on the right to the variable on the left.
- Add and Assign (+=): Adds the right operand to the left operand and assigns the result to the left operand.
- Subtract and Assign (-=): Subtracts the right operand from the left operand and assigns the result to the left operand.
- Multiply and Assign (=): Multiplies the left operand by the right operand and assigns the result to the left operand.
- Divide and Assign (/=): Divides the left operand by the right operand and assigns the result to the left operand.
- Modulus and Assign (%=): Takes the modulus of the left operand by the right operand and assigns the result to the left operand.

6. Unary Operators

Unary operators operate on a single operand.

- Unary Plus (+): Indicates a positive value.
- Unary Minus (-): Negates the value of the operand.
- Increment (++): Increases the value of the operand by 1.
- Decrement (--): Decreases the value of the operand by 1.

7. Ternary Operator

The ternary operator is a shorthand for the `if-else` statement.

- Syntax: `condition ? expression1 : expression2;`
- If the condition is true, `expression1` is executed; otherwise, `expression2` is executed.

Practice Questions on Java Operators

Now that we have a basic understanding of the various operators in Java, let’s dive into some practice questions designed to help you test your knowledge.

Question 1: Arithmetic Operators

Write a Java program that accepts two integers and performs the following operations:

- Addition
- Subtraction
- Multiplication
- Division
- Modulus

Expected Output:

```
Enter first integer: 10

Enter second integer: 3

Addition: 13

Subtraction: 7

Multiplication: 30

Division: 3

Modulus: 1
```

Question 2: Relational Operators

Given two integers, a and b, write a Java program to check:

- If a is equal to b.
- If a is not equal to b.
- If a is greater than b.
- If a is less than or equal to b.

Expected Output:

If a = 5 and b = 10:

```
a == b: false

a != b: true

a > b: false

a <= b: true
```

Question 3: Logical Operators

Write a Java program that checks if a number is both even and greater than 10.

Expected Output:

If the number is 12:

```
The number is even and greater than 10.
```

If the number is 9:

```
The number is not even or not greater than 10.
```

Question 4: Bitwise Operators

Create a Java program that demonstrates the use of bitwise operators with two integers. Display the results of AND, OR, XOR, and NOT operations.

Expected Output:

If a = 5 (0101) and b = 3 (0011):

```
Bitwise AND: 1 (0001)
Bitwise OR: 7 (0111)
Bitwise XOR: 6 (0110)
Bitwise NOT of a: -6 (in binary: 1010)
```

Question 5: Assignment Operators

Write a program that initializes an integer variable and demonstrates the use of various assignment operators.

Expected Output:

If the starting value is 5:

```
Initial value: 5
After += 3: 8
After -= 2: 6
After = 2: 12
After /= 3: 4
After %= 3: 1
```

Question 6: Ternary Operator

Implement a program that uses the ternary operator to determine if a number is positive, negative, or zero.

Expected Output:

If the number is -5:

```
The number is negative.
```

If the number is 0:

```
The number is zero.
```

If the number is 7:

```
The number is positive.
```

Conclusion

Practicing with Java operators is a crucial step in becoming proficient in Java programming. The questions outlined above cover a range of operator types and will help solidify your understanding of how to use them in various programming scenarios. By regularly testing yourself with these practice questions, you'll develop a strong foundation in Java operators, enabling you to tackle more complex programming challenges confidently. Happy coding!

Frequently Asked Questions

What is the output of the expression 5 + 10 2?

The output is 25 because multiplication has a higher precedence than addition.

How does the ++ operator work in Java?

The ++ operator increments a variable by 1. It can be used as a prefix (++x) or postfix (x++), affecting the order of operation.

What will be the result of the expression true && false || true?

The result will be true because the logical AND (&&) evaluates first, and true || false evaluates to true.

What is the difference between == and .equals() in Java?

The == operator checks for reference equality (same object), while .equals() checks for value equality (same content).

What will be the output of the following code: int a = 10; int b = 20;

System.out.println(a > b ? a : b);?

The output will be 20 because the ternary operator returns the second operand if the condition (a > b) is false.

Explain the use of the conditional (ternary) operator in Java.

The conditional operator (?:) evaluates a boolean expression and returns one of two values based on the result. For example: result = condition ? value1 : value2.

What is the output of the expression 10 % 3?

The output is 1 because the modulus operator (%) returns the remainder of the division of 10 by 3.

How do you perform bitwise AND operation on two integers in Java?

You can perform a bitwise AND operation using the & operator. For example: int result = a & b;.

What will happen if you use the / operator on two integers in Java?

Using the / operator on two integers performs integer division, meaning it will return the quotient without the remainder.

Find other PDF article:
https://soc.up.edu.ph/31-click/pdf?dataid=JWo26-2775&title=how-to-write-a-work-instruction-document.pdf

[Java Operators Practice Questions]()

□□ *Java* □□□□□ - □□
□□□□□ Java□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□2025□Java□□□□□ - □□
Jan 6, 2025 · Java□□□IT□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□java□□□□□□□30%□□□□□java□□□□□□□

Java□□□□□-CSDN□□□
Dec 30, 2024 · □□□□Java□□□□□□□□□□□□□□□Java□□□□□□□2023□□□□□□□□Java□□□□□□□□□□□□Java□□□□□□ □□□□□

□□□□□□□□□□□□□ …

Java LTS□□□□□□ - □□
Java LTS□□ (□□□□□□□)□□Bug□□□□□□□□□□□□
□Java LTS□□ …

Java□□-CSDN□□□
CSDNJava□□,Java□□,□□□□□□□□□□□□□□□□□□

Java□□□□□□□□□2024□□□□□□□ - □□
Java□□□□□□□□□ 2024□□□□□□□ □□SpringCloudAlibaba□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□
□□Java□□□□□□□□… □ …

Java□□□□□□□□□□ - □□
1 □□□Java□□□□□□□spring boot□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 2 □□1□□□□□JavaEE□□□□
□□□□□□□□ …

A Java Exception has occurred.□□□□□…-CSDN□□
Feb 7, 2010 · □□□□□□□□□"a java exception has occurred"□□□□□ □□□□□1.7□□□jdk□□□1.6□□□jdk□□□□□□□□□
□□□jdk□□□□ □jdk□□□eclipse□□□□□ …

□□!!! JDK□□□□□!-CSDN□□
Jun 2, 2014 · □□□□□CSDN□□□□□□!!! JDK□□□□□!□□□□□□□□□□□□□□□□Java SE□□□□□□□□□□□CSDN□□□

Spring Boot□□Redis□Lettuce□□□□□□□□□□□□□□□□ …
Apr 13, 2019 · □□□□□CSDN□□□□Spring Boot□□Redis□Lettuce□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□Java□□□□□□□□□□CSDN□□□

□□ *Java* □□□□□ - □□
□□□□□ Java□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□2025□Java□□□□□ - □□
Jan 6, 2025 · Java□□□IT□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□java□□□□□□□30%□□□□□java□□□□□□□

Java□□□□□-CSDN□□□
Dec 30, 2024 · □□□□Java□□□□□□□□□□□□□Java□□□□□□□2023□□□□□□□□Java□□□□□□□□□□□□Java□□□□□□ □□□□□
□□□□□□□□□□□□□ …

Java LTS□□□□□□ - □□
Java LTS□□ (□□□□□□□)□□Bug□□□□□□□□□□□□
□Java LTS□□ …

Java□□-CSDN□□□
CSDNJava□□,Java□□,□□□□□□□□□□□□□□□□□□

Java□□□□□□□□□2024□□□□□□□ - □□
Java□□□□□□□□□ 2024□□□□□□□ □□SpringCloudAlibaba□□□□□□□□□□□□□RocketMQ□□□□□□□□□□□□□□□□□□□□□□
□□Java□□□□□□□□… □ …

Java□□□□□□□□□□ - □□
1 □□□Java□□□□□□□spring boot□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 2 □□1□□□□□JavaEE□□□□

□□□□□□□□ …

A Java Exception has occurred.□□□□□…-CSDN□□
Feb 7, 2010 · □□□□□□□□□"a java exception has occurred"□□□□□ □□□□□1.7□□□jdk□□□1.6□□□jdk□□□□□□□□ □□□jdk□□□□□ □jdk□□□eclipse□□□□□ …

□□!!! JDK□□□□□!-CSDN□□
Jun 2, 2014 · □□□□□CSDN□□□□□□!!! JDK□□□□□!□□□□□□□□□□□□□□□□Java SE□□□□□□□□□□CSDN□□□

Spring Boot□□Redis□Lettuce□□□□□□□□□□□□□□□□ …
Apr 13, 2019 · □□□□□CSDN□□□□Spring Boot□□Redis□Lettuce□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Java□□□□□□□□□□CSDN□□□

Boost your Java skills with our curated list of Java operators practice questions. Test your knowledge and master operators today! Learn more now!

Back to Home