Name: ________________________

Java Math Operators

What are Math Operators?

Math operators are special symbols that tell Java to perform mathematical calculations. They work just like the math symbols you use in regular math class, but with a few differences to work with computer keyboards and programming rules.

Simple Analogy: Math Operators are Like Calculator Buttons

Think of Java math operators like buttons on a calculator:

- **Addition (+)** = The "+" button - adds numbers together
- **Subtraction (-)** = The "-" button - takes one number away from another
- ✖ **Multiplication (*)** = The "×" button - but we use * on keyboards
- **Division (/)** = The "÷" button - but we use / on keyboards
- **Remainder (%)** = Special button that gives you what's left over after division

Just like a calculator, Java follows the order of operations (PEMDAS) and gives you exact results!

Example: 5 + 3 × 2 = 11 (not 16!)
Multiplication happens first, then addition

The Five Basic Math Operators

Operator	Name	What It Does	Example	Result
+	Addition	Adds two numbers	10 + 3	13
-	Subtraction	Subtracts second number from first	10 - 3	7
*	Multiplication	Multiplies two numbers	10 * 3	30
/	Division	Divides first number by second	10 / 3	3.333... or 3 (depends on data type)
%	Modulus (Remainder)	Gives remainder after division	10 % 3	1 (because 10 ÷ 3 = 3 remainder 1)

Example 1: Basic Arithmetic

```
int a = 15;
int b = 4;

int sum = a + b; // Addition
int difference = a - b; // Subtraction
int product = a * b; // Multiplication
int quotient = a / b; // Division
int remainder = a % b; // Modulus

System.out.println("15 + 4 = " + sum); // 19
System.out.println("15 - 4 = " + difference); // 11
System.out.println("15 * 4 = " + product); // 60
System.out.println("15 / 4 = " + quotient); // 3 (integer division!)
System.out.println("15 % 4 = " + remainder); // 3
```

```
Output:
15 + 4 = 19
15 - 4 = 11
15 * 4 = 60
15 / 4 = 3
15 % 4 = 3
```

⚠ Important: Integer vs Decimal Division

Integer Division (int / int)

```
int a = 10;
int b = 3;
int result = a / b;

System.out.println(result); // 3 (not 3.33!)
```

Result: Java throws away the decimal part!

Decimal Division (double / double)

```
double a = 10.0;
double b = 3.0;
double result = a / b;

System.out.println(result); // 3.333333333...
```

Result: Java keeps the decimal part!

Rule: If you want decimal results, make sure at least one number is a double or float!

Example 2: Understanding Modulus (%) Operator

```
// Modulus gives you the "leftover" after division
System.out.println(17 % 5); // 2 (because 17 ÷ 5 = 3 remainder 2)
System.out.println(20 % 4); // 0 (because 20 ÷ 4 = 5 remainder 0)
System.out.println(7 % 3); // 1 (because 7 ÷ 3 = 2 remainder 1)

// Common uses of modulus:
int number = 23;
if (number % 2 == 0) {
  System.out.println("Even number");
} else {
    System.out.println("Odd number"); // This will print
}
```

Modulus is useful for:

- Checking if a number is even or odd
- Making numbers "wrap around" (like clock arithmetic)
- Finding patterns that repeat

Order of Operations (PEMDAS)

Java follows the same order of operations as math class:

Parentheses → **E**xponents → **M**ultiplication/Division → **A**ddition/Subtraction
(Left to right for operations of equal priority)

Example 3: Order of Operations in Action

```
int result1 = 2 + 3 * 4; // 14 (not 20!)
int result2 = (2 + 3) * 4; // 20
int result3 = 10 - 6 / 2 + 1; // 8
int result4 = 10 - (6 / 2 + 1); // 6

System.out.println("2 + 3 * 4 = " + result1);
System.out.println("(2 + 3) * 4 = " + result2);
System.out.println("10 - 6 / 2 + 1 = " + result3);
System.out.println("10 - (6 / 2 + 1) = " + result4);
```

Step-by-step for result3:

1. 10 - 6 / 2 + 1
2. 10 - 3 + 1 (division first)
3. 7 + 1 (left to right for same priority)
4. 8

Shorthand Assignment Operators

Java provides shortcuts for common operations where you modify a variable:

Shorthand	Long Form	What It Does	Example
+=	x = x + value	Add to variable	score += 10
-=	x = x - value	Subtract from variable	lives -= 1
*=	x = x * value	Multiply variable	price *= 1.1
/=	x = x / value	Divide variable	total /= 2
%=	x = x % value	Modulus on variable	index %= 10

Example 4: Using Shorthand Operators

```
int score = 100;
```

```
double price = 50.0;
int level = 1;
System.out.println("Initial score: " + score); // 100

score += 25; // Same as: score = score + 25
System.out.println("After bonus: " + score); // 125

score -= 10; // Same as: score = score - 10
System.out.println("After penalty: " + score); // 115

price *= 1.08; // Add 8% tax
System.out.println("Price with tax: $" + price); // $54.0
level *= 2; // Double the level
System.out.println("New level: " + level); // 2
```

Increment and Decrement Operators

Special shortcuts for adding or subtracting 1:

Operator	Name	What It Does	Same As
++	Increment	Add 1 to variable	x = x + 1 or x += 1
--	Decrement	Subtract 1 from variable	x = x - 1 or x -= 1

Example 5: Increment and Decrement

```
int counter = 5;
int lives = 3;
System.out.println("Counter: " + counter); // 5
counter++; // Add 1
System.out.println("Counter: " + counter); // 6

System.out.println("Lives: " + lives); // 3
lives--; // Subtract 1
System.out.println("Lives: " + lives); // 2
// Common in loops:
for (int i = 0; i < 5; i++) { // i++ adds 1 each time
  System.out.println("Loop iteration: " + i);
}
```

Example 6: Real-World Calculator Program

```
import java.util.Scanner;
public class SimpleCalculator {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.println("=== Simple Calculator ===");
    System.out.print("Enter first number: ");
    double num1 = input.nextDouble();

    System.out.print("Enter operator (+, -, *, /, %): ");
    String operator = input.next();

    System.out.print("Enter second number: ");
    double num2 = input.nextDouble();
    double result = 0;
    boolean validOperation = true;
    if (operator.equals("+")) {
      result = num1 + num2;
    } else if (operator.equals("-")) {
      result = num1 - num2;
    } else if (operator.equals("*")) {
      result = num1 * num2;
    } else if (operator.equals("/")) {
      if (num2 != 0) {
        result = num1 / num2;
      } else {
        System.out.println("Error: Cannot divide by zero!");
        validOperation = false;
      }
    } else if (operator.equals("%")) {
      result = num1 % num2;
    } else {
      System.out.println("Error: Invalid operator!");
      validOperation = false;
    }

    if (validOperation) {
      System.out.println(num1 + " " + operator + " " + num2 + " = " + result);
    }
    input.close();
  }
}
```

Example 7: Practical Math Applications

```
// Calculate compound interest
double principal = 1000.0; // Initial amount
double rate = 0.05; // 5% interest rate
int years = 3; // 3 years

double amount = principal;
for (int year = 1; year <= years; year++) {
  amount *= (1 + rate); // Multiply by 1.05 each year
  System.out.println("Year " + year + ": $" + amount);
}

// Convert temperature
double fahrenheit = 98.6;
double celsius = (fahrenheit - 32) * 5 / 9;
System.out.println(fahrenheit + "°F = " + celsius" + "°C");

// Calculate pizza slices per person
int totalSlices = 24;
int people = 7;
int slicesPerPerson = totalSlices / people;
int leftoverSlices = totalSlices % people;

System.out.println("Each person gets " + slicesPerPerson + " slices");
System.out.println("Leftover slices: " + leftoverSlices);
```

Common Math Operator Mistakes

Common Mistakes

```
// Integer division loses decimals
int result = 5 / 2; // 2, not 2.5!
// Division by zero crashes program
int x = 10 / 0; // ERROR!
// Wrong operator precedence
int y = 2 + 3 * 4; // 14, not 20!

// Mixing up modulus
if (x % 2 == 1) { // Check for odd
// Should be x % 2 != 0
}
```

Correct Solutions

```
// Use double for decimal results
double result = 5.0 / 2.0; // 2.5

// Always check before dividing
if (divisor != 0) {
  int x = 10 / divisor;
}
// Use parentheses when in doubt
int y = (2 + 3) * 4; // 20

// Proper even/odd check
if (x % 2 == 0) { // Even
// x % 2 gives 0 or 1
}
```

Math Operator Best Practices

Do This:

- **Use parentheses** to make order clear
- **Use doubles** when you need decimal results
- **Check for division by zero** before dividing
- **Use meaningful variable names**
- **Use shorthand operators** (+=, -=, etc.) for cleaner code
- **Comment complex calculations**

Don't Do This:

- **Assume integer division** gives decimals
- **Ignore operator precedence**
- **Divide by zero** without checking
- **Use confusing variable names** (a, b, c)
- **Write overly long expressions** without parentheses
- **Mix data types** without understanding results

Your Turn: Write Your Own Definition

What are math operators in Java? How would you explain them to a friend?

Write your definition in your own words:

Calculate the following expressions by hand, then write what Java would output:

Expression	Your Answer
15 / 4 (with int variables)	__________
15.0 / 4.0	__________
17 % 5	__________
2 + 3 * 4	__________
(2 + 3) * 4	__________

Explain when you would use the modulus (%) operator:

Write a real-world example where you would use each operator:

Addition (+): ____________________________________

Subtraction (-): ________________________________

Multiplication (*): _____________________________

Division (/): ___________________________________

Modulus (%): ___________________________________