1. Arithmetic & Assignment Operators

Q1: Write a program to swap two numbers without using a third variable and without using arithmetic operators like + or - .

Hint : Use bitwise XOR ^ operator.

Code:

class SwapUsingXOR {

public static void main(String[] args) {

int a = 5, b = 10;

System.out.println("Before swapping: a = " + a + ", b = " + b);

a = a ^ b;

b = a ^ b;

a = a ^ b;

System.out.println("After swapping: a = " + a + ", b = " + b);

}

}

OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.

Q2: Write a program to check whether a given number is even or odd using only bitwise operators .

Hint : Use n & 1 to check.

CODE:

class EvenOddUsingBitwise {

public static void main(String[] args) {

int n = 7; // Change this value to test

if ((n & 1) == 0) {

System.out.println(n + " is Even");

} else {

System.out.println(n + " is Odd");

}

}

}

OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.

Q3: Implement a program that calculates the sum of digits of an integer using modulus ( % ) and division ( / ) operators .

CODE:

class SumOfDigits {

public static void main(String[] args) {

int num = 12345; // Change this value to test

int sum = 0;

while (num > 0) {

sum += num % 10; // Extract last digit

num /= 10; // Remove last digit

}

System.out.println("Sum of digits: " + sum);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q4: Write a program to find whether a given number is divisible by 3 without using the modulus ( % ) or division ( / ) operators. Hint : Use subtraction and bitwise shifts .

CODE:

class DivisibilityByThree {

public static boolean isDivisibleBy3(int num) {

num = Math.abs(num); // Handle negative numbers

while (num > 3) {

int sum = 0;

while (num > 0) {

sum += num & 1; // Extract last bit

num >>= 1; // Right shift (divide by 2)

}

num = sum;

}

return (num == 0 || num == 3);

}

public static void main(String[] args) {

int num = 27; // Change this value to test

System.out.println(num + (isDivisibleBy3(num) ? " is" : " is not") + " divisible by 3");

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q5: Write a Java program to swap two numbers using the += and -= operators only.

CODE:

class SwapUsingPlusMinus {

public static void main(String[] args) {

int a = 5, b = 10;

System.out.println("Before swapping: a = " + a + ", b = " + b);

a += b; // a = a + b

b = a - b; // b = (a + b) - b -> original a

a -= b; // a = (a + b) - a -> original b

System.out.println("After swapping: a = " + a + ", b = " + b);

}

}

OUTPUT:A screenshot of a computer

AI-generated content may be incorrect.

1. Relational & Logical Operators

Q6: Write a program to find the largest of three numbers using only the ternary operator ( ? : ) .

CODE:

class LargestUsingTernary {

public static void main(String[] args) {

int a = 25, b = 40, c = 15;

int largest = (a > b) ? (a > c ? a : c) : (b > c ? b : c);

System.out.println("The largest number is: " + largest);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q7: Implement a Java program that checks whether a given year is a leap year or not using logical ( && , || ) operators .

CODE:

class LeapYearCheck {

public static void main(String[] args) {

int year = 2024; // Change this value to test

boolean isLeap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

System.out.println(year + " is " + (isLeap ? "a Leap Year" : "not a Leap Year"));

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q8: Write a program that takes three boolean inputs and prints true if at least two of them are true . Hint : Use logical operators ( && , || ).

CODE:

class AtLeastTwoTrue {

public static void main(String[] args) {

boolean a = true, b = false, c = true;

boolean result = (a && b) || (b && c) || (a && c);

System.out.println("At least two are true: " + result);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q9: Implement a Java program that checks if a number is within a specific range (20 to 50) without using if-else . Hint : Use logical AND ( && ) in a print statement .

CODE:

class NumberInRange {

public static void main(String[] args) {

int num = 35; // Change this value to test

System.out.println(num + " is in range (20-50): " + (num >= 20 && num <= 50));

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q10: Write a program to determine if a character is a vowel or a consonant using the ternary operator.

CODE:

class VowelOrConsonant {

public static void main(String[] args) {

char ch = 'e'; // Change this value to test

ch = Character.toLowerCase(ch);

String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') ? "Vowel" : "Consonant";

System.out.println(ch + " is a " + result);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

1. Bitwise Operators

Q11: Write a program to check if a given number is a power of 2 using bitwise operators. Hint : n & (n - 1) == 0 for positive numbers.

CODE:

class PowerOfTwo {

public static void main(String[] args) {

int n = 16; // Change this value to test

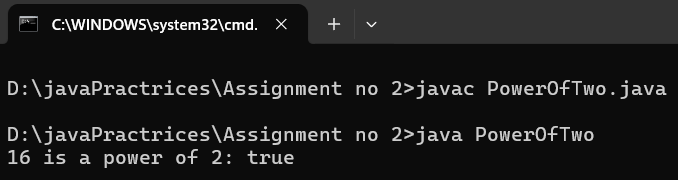
boolean isPowerOfTwo = (n > 0) && ((n & (n - 1)) == 0);

System.out.println(n + " is a power of 2: " + isPowerOfTwo);

}

}

OUTPUT:



Q12: Write a Java program to multiply a number by 8 without using \* or / operators. Hint : Use bitwise left shift ( << ).

CODE:

class MultiplyByEight {

public static void main(String[] args) {

int num = 5; // Change this value to test

int result = num << 3; // Left shift by 3 is equivalent to multiplying by 8

System.out.println(num + " multiplied by 8 is: " + result);

}

}

OUTPUT:

A screenshot of a phone

AI-generated content may be incorrect.

Q13: Implement a Java program to find the absolute value of an integer using bitwise operators. Hint : mask = num >> 31; abs = (num + mask) ^ mask;

CODE:

class AbsoluteValue {

public static void main(String[] args) {

int num = -15; // Change this value to test

int mask = num >> 31; // Extracts sign bit (0 for positive, -1 for negative)

int absValue = (num + mask) ^ mask;

System.out.println("Absolute value of " + num + " is: " + absValue);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q14: Write a program to count the number of 1s (set bits) in a binary representation of a number using bitwise operations. Hint : Use n & (n - 1) .

CODE:

class CountSetBits {

public static void main(String[] args) {

int num = 13; // Change this value to test

int count = 0;

while (num > 0) {

num = num & (num - 1); // Clears the rightmost set bit

count++;

}

System.out.println("Number of set bits: " + count);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q15: Implement a program to swap odd and even bits of a number using bitwise operators. Hint : Use masks: (x & 0xAAAAAAAA) >> 1 | (x & 0x55555555) << 1

CODE:

class SwapOddEvenBits {

public static void main(String[] args) {

int x = 23; // Change this value to test

int evenBits = x & 0xAAAAAAAA; // Mask for even bits

int oddBits = x & 0x55555555; // Mask for odd bits

int result = (evenBits >> 1) | (oddBits << 1);

System.out.println("Number after swapping odd and even bits: " + result);

}

}

OUTPUT:

A screenshot of a phone

AI-generated content may be incorrect.

1. Ternary Operator Challenges

Q16: Write a program that determines whether a given number is positive, negative, or zero using only the ternary operator .

CODE:

class NumberCheck {

public static void main(String[] args) {

int num = -5; // Change this value to test

String result = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";

System.out.println("The number is: " + result);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q17: Implement a Java program that finds the minimum of four numbers using nested ternary operators.

CODE:

class MinimumOfFour {

public static void main(String[] args) {

int a = 15, b = 7, c = 20, d = 5; // Change values to test

int min = (a < b) ? ((a < c) ? ((a < d) ? a : d) : (c < d) ? c : d)

: ((b < c) ? ((b < d) ? b : d) : (c < d) ? c : d);

System.out.println("Minimum number is: " + min);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q18: Given a student’s percentage, print “Pass” if the percentage is 40 or above; otherwise, print “Fail” , using only the ternary operator.

CODE:

class PassFailCheck {

public static void main(String[] args) {

int percentage = 38; // Change this value to test

String result = (percentage >= 40) ? "Pass" : "Fail";

System.out.println("Result: " + result);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q19: Write a Java program that checks whether a character is uppercase, lowercase, or not a letter using only the ternary operator.

CODE:

class CharacterCheck {

public static void main(String[] args) {

char ch = 'A'; // Change this value to test

String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase" :

(ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a Letter";

System.out.println("Character type: " + result);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Q20: Implement a Java program that returns the absolute value of a given number using the ternary operator (without using Math.abs() ).

CODE:

class AbsoluteValueTernary {

public static void main(String[] args) {

int num = -25; // Change this value to test

int absValue = (num < 0) ? -num : num;

System.out.println("Absolute value: " + absValue);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.