**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.

Ans :

public class Operators {

public static void main(String[] args) {

int a = 5, b = 7;

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

a = a ^ b;

b = a ^ b;

a = a ^ b;

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

}

}

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

Hint: Use n & 1 to check.

Ans :

class EvenOdd {

public static void main(String[] args) {

int n = 11;

if ((n & 1) == 0)

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

else

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

}

}

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

Ans :

class SumOfDigits {

public static void main(String[] args) {

int num = 12345, sum = 0;

while (num != 0) {

sum += num % 10;

num /= 10;

}

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

}

}

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.

Ans :

class DivisibleByThree {

public static void main(String[] args) {

int num = 27;

while (num > 3) {

num -= 3;

}

if (num == 0 || num == 3)

System.out.println("Divisible by 3");

else

System.out.println("Not divisible by 3");

}

}

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

Ans :

class SwapNumbers {

public static void main(String[] args) {

int x = 10, y = 20;

System.out.println("Before Swap: x = " + x + ", y = " + y);

x += y; // x = x + y

y = x - y; // y = (x + y) - y => y = x

x -= y; // x = (x + y) - x => x = y

System.out.println("After Swap: x = " + x + ", y = " + y);

}

}

**2. Relational & Logical Operators**

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

Ans :

class LargestNumber {

public static void main(String[] args) {

int a = 10, b = 20, c = 15;

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

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

}

}

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

Ans :

class LeapYear {

public static void main(String[] args) {

int year = 2024;

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

System.out.println(year + " is Leap Year: " + isLeap);

}

}

Q8: Write a program that takes three boolean inputs and prints true if at least two of them are true.

Hint: Use logical operators (&&, ||).

Ans :

class TwoTrue {

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);

}

}

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.

Ans :

class NumberInRange {

public static void main(String[] args) {

int num = 30;

System.out.println((num >= 20 && num <= 50) ? "Within range" : "Out of range");

}

}

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

Ans :

class VowelConsonant {

public static void main(String[] args) {

char ch = 'a';

String result = (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')

? "Vowel" : "Consonant";

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

}

}

**3. 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.

Ans :

class PowerOfTwo {

public static void main(String[] args) {

int n = 16;

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

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

}

}

Q12: Write a Java program to multiply a number by 8 without using or/ operators. \*

Hint: Use bitwise left shift (<<).

Ans :

class MultiplyByEight {

public static void main(String[] args) {

int num = 5;

int result = num << 3; // num \* 8

System.out.println("5 multiplied by 8 is: " + result);

}

}

Q13: Implement a Java program to find the absolute value of an integer using bitwise operators.

Hint: mask = num >> 31; abs = ww (num + mask) A mask;

Ans:

class AbsoluteValue {

public static void main(String[] args) {

int num = -23;

int mask = num >> 31;

int abs = (num + mask) ^ mask;

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

}

}

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)

Ans :

class CountSetBits {

public static void main(String[] args) {

int n = 29, count = 0;

while (n > 0) {

n = n & (n - 1);

count++;

}

System.out.println("Number of 1s: " + count);

}

}

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

Ans :

class SwapOddEvenBits {

public static void main(String[] args) {

int x = 23;

int result = ((x & 0xAAAAAAAA) >> 1) | ((x & 0x55555555) << 1);

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

}

}

**4. Ternary Operator Challenges**

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

Ans :

class PositiveNegativeZero {

public static void main(String[] args) {

int num = -5;

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

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

}

}

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

Ans :

class MinimumOfFour {

public static void main(String[] args) {

int a = 10, b = 20, c = 5, d = 15;

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: " + min);

}

}

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

Ans :

class PassFail {

public static void main(String[] args) {

int percentage = 45;

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

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

}

}

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

Ans :

class CharacterType {

public static void main(String[] args) {

char ch = 'A';

String result = (ch >= 'A' && ch <= 'Z') ? "Uppercase" : (ch >= 'a' && ch <= 'z') ? "Lowercase" : "Not a Letter";

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

}

}

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

Ans :

class AbsoluteValueTernary {

public static void main(String[] args) {

int num = -30;

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

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

}

}

**5. Miscellaneous Operator Questions**

Q21: Write a program that increments a number without using + or ++ operators.

Hint: Use bitwise (~x)

Ans :

class IncrementNumber {

public static void main(String[] args) {

int num = 5;

num = -~num;

System.out.println("Incremented number: " + num);

}

}

Q22: Implement a calculator that takes two numbers and an operator (+, -, \*, /) as input and prints the result using only switch-case.

Ans :

class Calculator {

public static void main(String[] args) {

int a = 10, b = 5;

char operator = '+';

switch (operator) {

case '+': System.out.println("Result: " + (a + b)); break;

case '-': System.out.println("Result: " + (a - b)); break;

case '\*': System.out.println("Result: " + (a \* b)); break;

case '/': System.out.println("Result: " + (a / b)); break;

default: System.out.println("Invalid operator");

}

}

}

Q23: Given a number, find whether it is odd or even using the & bitwise operator and print the result without using if-else.

Ans :

class EvenOddBitwise {

public static void main(String[] args) {

int num = 14;

System.out.println((num & 1) == 0 ? "Even" : "Odd");

}

}

Q24: Write a program that prints all even numbers from 1 to 100 using only bitwise AND (&) and for loop.

Ans :

class EvenNumbers {

public static void main(String[] args) {

for (int i = 1; i <= 100; i++) {

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

System.out.print(i + " ");

}

}

}

}

Q25: Implement a program that reverses an integer number without using string conversion (StringBuilder or to toCharArray).

Hint: Use while (n!=0) { rev = rev 10+n%10; n = 10; } \*

Ans :

class ReverseNumber {

public static void main(String[] args) {

int num = 12345, rev = 0;

while (num != 0) {

rev = rev \* 10 + num % 10;

num /= 10;

}

System.out.println("Reversed number: " + rev);

}

}