Name:032\_Kaustubh\_Gade\_JH

1. Arithmetic & Assignment Operators

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

Program:

class Main {

public static void main(String[] args) {

int a=4,b=5;

System.out.println("Before Swap a & b is "+a+" and "+b);

a=a^b;

b=a^b;

a=a^b;

System.out.println("After Swap a & b is "+a+" and "+b);

}

}

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

Program:

public class EvenOddCheck {

public static void main(String[] args) {

int n = 7;

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 .

Program:

public class SumOfDigits {

public static void main(String[] args) {

int num = 12345;

int sum = 0;

while (num != 0) {

sum += num % 10; // Get the last digit

num /= 10; // Remove the last digit

}

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.

Program:

public class Main

{

public static void main(String[] args) {

int n=5,b;

b=n;

while(n>=3)

{

n=n-3;

}

if(n==0)

{

System.out.println(b+" is Divisble by 3");

}

else{

System.out.println(b+" is not Divisble by 3");

}

}

}

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

Program:

public class Main

{

public static void main(String[] args)

{

int a=5,b=10;

a=a+b;

b=a-b;

a=a-b;

}

}

1. Relational & Logical Operators

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

Program:

public class Main

{

public static void main(String[] args)

{

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

String result=(a>b&&a>c)? a +" is greater":(b>a&&b>c)?b+" is greater":c+" is greater";

System.out.println(result);

}

}

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

Program:

public class Main

{

public static void main(String[] args)

{

int year=2004;

if(year%4==0 && (year%100==0 || year%400!=0))

{

System.out.println("Leap Year");

}

else{

System.out.println(" Not Leap Year");

}

}

}

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

Program:

public class Main

{

public static void main(String[] args)

{

boolean a=false;

boolean b=false;

boolean c=true;

if((a&&b) || (a&&c)||(b&&c))

{

System.out.println("Is True");

}

else{

System.out.println("Not True");

}

}

}

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

Program:

public class Main

{

public static void main(String[] args)

{

int num=30;

System.out.println(num>=20 && num<=50);

}

}

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

Program:

public class Main

{

public static void main(String[] args) {

char word='f';

String result=(word=='a' || word=='e' || word=='i' || word=='o' || word=='u')?"Is vowel":"Not Vowel";

System.out.println(result);

}

}

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.

Program:-

public class Main

{

public static void main(String[] args) {

int n=17;

if(n>0 &&(n&(n-1))==0)

{

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

}

else{

System.out.println(n + " is not power of 2.");

}

}

}

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

Program:

public class Main

{

public static void main(String[] args) {

int n=5;

int result=n<<3;

System.out.println(n+" 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 = (num + mask) ^ mask;

Program:

public class AbsoluteValue {

public static int absoluteValue(int num) {

int mask = num >> 31;

return (num + mask) ^ mask;

}

public static void main(String[] args) {

int num = -15;

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

}

}

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)

Program:

public class CountSetBits {

public static int countSetBits(int n) {

int count = 0;

while (n > 0) {

n = n & (n - 1);

count++;

}

return count;

}

public static void main(String[] args) {

int n = 15;

System.out.println("Number of set bits in " + n + " is: " + countSetBits(n));

}

}

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

Program:

public class SwapOddEvenBits {

public static int swapBits(int x) {

int evenBits = x & 0x55555555; // 0101... (binary)

int oddBits = x & 0xAAAAAAAA; // 1010... (binary)

evenBits <<= 1;

oddBits >>= 1;

return (evenBits | oddBits);

}

public static void main(String[] args) {

int x = 23; // Binary: 10111

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

}

}

1. **Ternary Operator Challenges**

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

Program:

public class AbsoluteValue {

public static void main(String[] args) {

int num = -15;

int mask = num >> 31;

int abs = (num + mask) ^ mask;

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

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

Program:

public class MinimumOfFour {

public static void main(String[] args) {

int a = 10, b = 20, c = 5, d = 30;

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 value is: " + min);

}

}

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

Program:

public class PassFail {

public static void main(String[] args) {

double 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.

Program:

public class CharacterCheck {

public static void main(String[] args) {

char c = 'A';

String result = (Character.isUpperCase(c)) ? "Uppercase" :

(Character.isLowerCase(c)) ? "Lowercase" : "Not a letter";

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

}

}

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

Program:

public class AbsoluteValueUsingTernary {

public static void main(String[] args) {

int num = -15;

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

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

}

}

1. Miscellaneous Operator Questions

Q21: Write a program that increments a number without using + or ++ operators. Hint : Use bitwise - (~x)

Program:

public class IncrementWithoutPlus {

public static void main(String[] args) {

int num = 5;

int incrementedNum = ~(-num); // Using bitwise NOT operator

System.out.println("Incremented value: " + incrementedNum);

}

}

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

Program:

import java.util.Scanner;

public class Calculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first number: ");

int num1 = scanner.nextInt();

System.out.print("Enter second number: ");

int num2 = scanner.nextInt();

System.out.print("Enter operator (+, -, \*, /): ");

char operator = scanner.next().charAt(0);

int result = 0;

switch (operator) {

case '+': result = num1 + num2; break;

case '-': result = num1 - num2; break;

case '\*': result = num1 \* num2; break;

case '/': result = num1 / num2; break;

default: result = 0; break;

}

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

}

}

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

Program:

public class OddEvenUsingBitwise {

public static void main(String[] args) {

int num = 10;

String result = (num & 1) == 0 ? "Even" : "Odd";

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

}

}

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

Program:

public class EvenNumbers {

public static void main(String[] args) {

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

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

System.out.println(i);

}

}

}

}

Q25: Implement a program that reverses an integer number without using string conversion ( StringBuilder or toCharArray ). Hint : Use while(n!=0) { rev = rev \* 10 + n % 10; n /= 10; }

Program:

public class ReverseInteger {

public static void main(String[] args) {

int num = 12345;

int rev = 0;

while (num != 0) {

rev = rev \* 10 + num % 10;

num /= 10;

}

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

}

}