**1. Arithmetic Operators**

**Multiplication:** Write a Java program that multiplies two given integers and prints the result.

java

Copy code

public class Multiplication {

public static void main(String[] args) {

int a = 5;

int b = 7;

int result = a \* b;

System.out.println("The product of " + a + " and " + b + " is " + result);

}

}

**Division:** Write a Java program that divides one integer by another and prints the quotient.

java

Copy code

public class Division {

public static void main(String[] args) {

int a = 10;

int b = 2;

int result = a / b;

System.out.println("The quotient of " + a + " divided by " + b + " is " + result);

}

}

**Modulo:** Write a Java program that finds the remainder of the division of two integers and prints it.

java

Copy code

public class Modulo {

public static void main(String[] args) {

int a = 10;

int b = 3;

int result = a % b;

System.out.println("The remainder of " + a + " divided by " + b + " is " + result);

}

}

**Addition:** Write a Java program that adds two integers and prints the result.

java

Copy code

public class Addition {

public static void main(String[] args) {

int a = 5;

int b = 7;

int result = a + b;

System.out.println("The sum of " + a + " and " + b + " is " + result);

}

}

**Subtraction:** Write a Java program that subtracts one integer from another and prints the result.

java

Copy code

public class Subtraction {

public static void main(String[] args) {

int a = 10;

int b = 3;

int result = a - b;

System.out.println("The difference between " + a + " and " + b + " is " + result);

}

}

**2. Unary Operators**

**Unary minus:** Write a Java program that negates an integer and prints the result.

java

Copy code

public class UnaryMinus {

public static void main(String[] args) {

int a = 10;

int result = -a;

System.out.println("The negation of " + a + " is " + result);

}

}

**Unary plus:** Write a Java program that indicates a positive value and prints it.

java

Copy code

public class UnaryPlus {

public static void main(String[] args) {

int a = 10;

int result = +a;

System.out.println("The positive value of " + a + " is " + result);

}

}

**Increment:** Write a Java program that demonstrates post-increment and pre-increment operations.

java

Copy code

public class Increment {

public static void main(String[] args) {

int a = 10;

System.out.println("Post-increment: " + (a++)); // Prints 10, then a becomes 11

System.out.println("Pre-increment: " + (++a)); // a becomes 12, then prints 12

}

}

**Decrement:** Write a Java program that demonstrates post-decrement and pre-decrement operations.

java

Copy code

public class Decrement {

public static void main(String[] args) {

int a = 10;

System.out.println("Post-decrement: " + (a--)); // Prints 10, then a becomes 9

System.out.println("Pre-decrement: " + (--a)); // a becomes 8, then prints 8

}

}

**Logical NOT:** Write a Java program that inverts the value of a boolean expression and prints the result.

java

Copy code

public class LogicalNot {

public static void main(String[] args) {

boolean isTrue = true;

boolean result = !isTrue;

System.out.println("The negation of " + isTrue + " is " + result);

}

}

**3. Assignment Operators**

**Basic assignment:** Write a Java program that assigns a value to an integer variable and prints it.

java

Copy code

public class BasicAssignment {

public static void main(String[] args) {

int f = 7;

System.out.println("The value of f is " + f);

}

}

**Addition assignment:** Write a Java program that adds a value to an integer variable using addition assignment and prints the result.

java

Copy code

public class AdditionAssignment {

public static void main(String[] args) {

int f = 7;

f += 3;

System.out.println("The value of f after addition assignment is " + f);

}

}

**Subtraction assignment:** Write a Java program that subtracts a value from an integer variable using subtraction assignment and prints the result.

java

Copy code

public class SubtractionAssignment {

public static void main(String[] args) {

int f = 7;

f -= 3;

System.out.println("The value of f after subtraction assignment is " + f);

}

}

**Multiplication assignment:** Write a Java program that multiplies an integer variable by a value using multiplication assignment and prints the result.

java

Copy code

public class MultiplicationAssignment {

public static void main(String[] args) {

int f = 7;

f \*= 3;

System.out.println("The value of f after multiplication assignment is " + f);

}

}

**Division assignment:** Write a Java program that divides an integer variable by a value using division assignment and prints the result.

java

Copy code

public class DivisionAssignment {

public static void main(String[] args) {

int f = 7;

f /= 3;

System.out.println("The value of f after division assignment is " + f);

}

}

**Modulo assignment:** Write a Java program that finds the remainder of the division of an integer variable by a value using modulo assignment and prints the result.

java

Copy code

public class ModuloAssignment {

public static void main(String[] args) {

int f = 7;

f %= 3;

System.out.println("The value of f after modulo assignment is " + f);

}

}

**4. Relational Operators**

**Equal to:** Write a Java program that checks if two integers are equal and prints the result.

java

Copy code

public class EqualTo {

public static void main(String[] args) {

int a = 5;

int b = 7;

boolean result = (a == b);

System.out.println("Is " + a + " equal to " + b + "? " + result);

}

}

**Not equal to:** Write a Java program that checks if two integers are not equal and prints the result.

java

Copy code

public class NotEqualTo {

public static void main(String[] args) {

int a = 5;

int b = 7;

boolean result = (a != b);

System.out.println("Is " + a + " not equal to " + b + "? " + result);

}

}

**Less than:** Write a Java program that checks if one integer is less than another and prints the result.

java

Copy code

public class LessThan {

public static void main(String[] args) {

int a = 5;

int b = 7;

boolean result = (a < b);

System.out.println("Is " + a + " less than " + b + "? " + result);

}

}

**Less than or equal to:** Write a Java program that checks if one integer is less than or equal to another and prints the result.

java

Copy code

public class LessThanOrEqualTo {

public static void main(String[] args) {

int a = 7;

int b = 7;

boolean result = (a <= b);

System.out.println("Is " + a + " less than or equal to " + b + "? " + result);

}

}

**Greater than:** Write a Java program that checks if one integer is greater than another and prints the result.

java

Copy code

public class GreaterThan {

public static void main(String[] args) {

int a = 9;

int b = 7;

boolean result = (a > b);

System.out.println("Is " + a + " greater than " + b + "? " + result);

}

}

**Greater than or equal to:** Write a Java program that checks if one integer is greater than or equal to another and prints the result.

java

Copy code

public class GreaterThanOrEqualTo {

public static void main(String[] args) {

int a = 7;

int b = 7;

boolean result = (a >= b);

System.out.println("Is " + a + " greater than or equal to " + b + "? " + result);

}

}

**5. Logical Operators**

**Logical AND:** Write a Java program that checks if two boolean values are both true and prints the result.

java

Copy code

public class LogicalAND {

public static void main(String[] args) {

boolean x = true;

boolean y = false;

boolean result = (x && y);

System.out.println("Are both x and y true? " + result);

}

}

**Logical OR (||) Operator:**

Checks whether a given integer is either divisible by 3 or divisible by 5.

**Logical NOT (!) Operator:**

Determines whether a given year is a leap year. Use the logical NOT operator to check if the year is not divisible by 4 or is divisible by 100 but not by 400.

**Bitwise Operator:**

**Bitwise AND (&) Operator:**

Takes two positive integers as input and returns their bitwise AND result.

**Bitwise OR (|) Operator:**

Computes the bitwise OR of two binary numbers represented as strings (e.g., “1101” OR “1010” should yield “1111”).

**Bitwise XOR (^) Operator:**

Calculates the bitwise XOR of two integers and returns the result.

**Bitwise Complement (~) Operator:**

Takes an integer as input and prints its bitwise complement (i.e., flipping all the bits).

**Shift Operator:**

**Left Shift (<<) Operator:**

Design a Kotlin function that shifts the bits of an integer to the left by a specified number of positions.

**Signed Right Shift (>>) Operator:**

Write a Swift program that performs a signed right shift on a given integer by a user-defined number of bits.

**Unsigned Right Shift (>>>) Operator:**

Develop a Rust function that performs an unsigned right shift on a 32-bit unsigned integer by a specified number of positions.