1. Declare and initialize all primitive data types in Java:

java

Copy

Edit

public class PrimitiveTypes {

public static void main(String[] args) {

byte b = 10;

short s = 20;

int i = 30;

long l = 40L;

float f = 50.5f;

double d = 60.6;

char c = 'A';

boolean bool = true;

System.out.println("byte: " + b);

System.out.println("short: " + s);

System.out.println("int: " + i);

System.out.println("long: " + l);

System.out.println("float: " + f);

System.out.println("double: " + d);

System.out.println("char: " + c);

System.out.println("boolean: " + bool);

}

}

2>

public class IntToDouble {

public static void main(String[] args) {

int num = 42;

double converted = num; // Implicit conversion

System.out.println("Integer value: " + num);

System.out.println("Converted double value: " + converted);

}

}

3>

public class DoubleToInt {

public static void main(String[] args) {

double num = 42.99;

int converted = (int) num; // Explicit conversion

System.out.println("Double value: " + num);

System.out.println("Converted int value: " + converted);

System.out.println("Data loss occurs as decimal part is truncated.");

}

}

4>

public class AverageCalculator {

public static void main(String[] args) {

int num1 = 10, num2 = 20, num3 = 30;

double average = (num1 + num2 + num3) / 3.0; // Typecasting

System.out.println("Average: " + average);

}

}

4>

public class LiteralsDemo {

public static void main(String[] args) {

int binary = 0b1010; // Binary (10)

int octal = 012; // Octal (10)

int hex = 0xA; // Hexadecimal (10)

float floatNum = 10.5f; // Floating-point

System.out.println("Binary: " + binary);

System.out.println("Octal: " + octal);

System.out.println("Hexadecimal: " + hex);

System.out.println("Floating-point: " + floatNum);

}

}

5>

public class LiteralsDemo {

public static void main(String[] args) {

int binary = 0b1010; // Binary (10)

int octal = 012; // Octal (10)

int hex = 0xA; // Hexadecimal (10)

float floatNum = 10.5f; // Floating-point

System.out.println("Binary: " + binary);

System.out.println("Octal: " + octal);

System.out.println("Hexadecimal: " + hex);

System.out.println("Floating-point: " + floatNum);

}

}

6>

public class CharLiterals {

public static void main(String[] args) {

char ch = 'A';

String str = "Hello";

System.out.println("Character: " + ch + " ASCII: " + (int) ch);

for (char c : str.toCharArray()) {

System.out.println("Character: " + c + " ASCII: " + (int) c);

}

}

}

7>

public class BooleanDemo {

public static void main(String[] args) {

boolean isJavaFun = true;

if (isJavaFun) {

System.out.println("Java is fun!");

} else {

System.out.println("Java is not fun!");

}

}

}

8>

public class ArithmeticOperations {

public static void main(String[] args) {

int a = 15, b = 4;

System.out.println("Addition: " + (a + b));

System.out.println("Subtraction: " + (a - b));

System.out.println("Multiplication: " + (a \* b));

System.out.println("Division: " + (a / b));

System.out.println("Modulus: " + (a % b));

}

}

9>

public class RelationalOperators {

public static void main(String[] args) {

int a = 10, b = 20;

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

System.out.println("a != b: " + (a != b));

System.out.println("a > b: " + (a > b));

System.out.println("a < b: " + (a < b));

System.out.println("a >= b: " + (a >= b));

System.out.println("a <= b: " + (a <= b));

}

}

10>

public class LogicalOperators {

public static void main(String[] args) {

int num = 8;

if (num > 0 && num % 2 == 0) {

System.out.println(num + " is positive and even.");

} else {

System.out.println(num + " is not positive or not even.");

}

}

}

11>

public class AssignmentOperators {

public static void main(String[] args) {

int x = 10;

x += 5; // x = x + 5

System.out.println("x after += : " + x);

x -= 3; // x = x - 3

System.out.println("x after -= : " + x);

x \*= 2; // x = x \* 2

System.out.println("x after \*= : " + x);

x /= 4; // x = x / 4

System.out.println("x after /= : " + x);

x %= 3; // x = x % 3

System.out.println("x after %= : " + x);

}

}