Sample programs

1. Byte :

class JavaExample {

public static void main(String[] args) {

byte num;

num = 113;

System.out.println(num);

}

}

1. Modify the value to 200 – to get type mismatch error
2. Short

class JavaExample {

public static void main(String[] args) {

short num;

num = 150;

System.out.println(num);

}

}

1. Long

class JavaExample {

public static void main(String[] args) {

long num = -12332252626L;

System.out.println(num);

}

}

1. Double

class JavaExample {

public static void main(String[] args) {

double num = -42937737.9d;

System.out.println(num);

}

}

1. Float

class JavaExample {

public static void main(String[] args) {

float num = 19.98f;

System.out.println(num);

}

}

1. Boolean

class JavaExample {

public static void main(String[] args) {

boolean b = false;

System.out.println(b);

}

}

1. Arithmetic Operators

public class ArithmeticOperatorDemo {

public static void main(String args[]) {

int num1 = 100;

int num2 = 20;

System.out.println("num1 + num2: " + (num1 + num2) );

System.out.println("num1 - num2: " + (num1 - num2) );

System.out.println("num1 \* num2: " + (num1 \* num2) );

System.out.println("num1 / num2: " + (num1 / num2) );

System.out.println("num1 % num2: " + (num1 % num2) );

}

}

1. Assignment Operators

public class AssignmentOperatorDemo {

public static void main(String args[]) {

int num1 = 10;

int num2 = 20;

num2 = num1;

System.out.println("= Output: "+num2);

num2 += num1;

System.out.println("+= Output: "+num2);

num2 -= num1;

System.out.println("-= Output: "+num2);

num2 \*= num1;

System.out.println("\*= Output: "+num2);

num2 /= num1;

System.out.println("/= Output: "+num2);

num2 %= num1;

System.out.println("%= Output: "+num2);

}

}

1. Auto increment & auto decrement Operators

public class AutoOperatorDemo {

public static void main(String args[]){

int num1=100;

int num2=200;

num1++;

num2--;

System.out.println("num1++ is: "+num1);

System.out.println("num2-- is: "+num2);

}

}

1. Relational Operators

public class RelationalOperatorDemo {

public static void main(String args[]) {

int num1 = 10;

int num2 = 50;

if (num1==num2) {

System.out.println("num1 and num2 are equal");

}

else{

System.out.println("num1 and num2 are not equal");

}

if( num1 != num2 ){

System.out.println("num1 and num2 are not equal");

}

else{

System.out.println("num1 and num2 are equal");

}

if( num1 > num2 ){

System.out.println("num1 is greater than num2");

}

else{

System.out.println("num1 is not greater than num2");

}

if( num1 >= num2 ){

System.out.println("num1 is greater than or equal to num2");

}

else{

System.out.println("num1 is less than num2");

}

if( num1 < num2 ){

System.out.println("num1 is less than num2");

}

else{

System.out.println("num1 is not less than num2");

}

if( num1 <= num2){

System.out.println("num1 is less than or equal to num2");

}

else{

System.out.println("num1 is greater than num2");

}

}

}

1. Logical Operators

public class LogicalOperatorDemo {

public static void main(String args[]) {

boolean b1 = true;

boolean b2 = false;

System.out.println("b1 && b2: " + (b1&&b2));

System.out.println("b1 || b2: " + (b1||b2));

System.out.println("!(b1 && b2): " + !(b1&&b2));

}

}

1. Bitwise Operator

public class BitwiseOperatorDemo {

public static void main(String args[]) {

int num1 = 11; /\* 11 = 00001011 \*/

int num2 = 22; /\* 22 = 00010110 \*/

int result = 0;

result = num1 & num2;

System.out.println("num1 & num2: "+result);

result = num1 | num2;

System.out.println("num1 | num2: "+result);

result = num1 ^ num2;

System.out.println("num1 ^ num2: "+result);

result = ~num1;

System.out.println("~num1: "+result);

result = num1 << 2;

System.out.println("num1 << 2: "+result); result = num1 >> 2;

System.out.println("num1 >> 2: "+result);

}

}

14 .Ternary Operator

public class TernaryOperatorDemo {

public static void main(String args[]) {

int num1, num2;

num1 = 25;

/\* num1 is not equal to 10 that's why

\* the second value after colon is assigned

\* to the variable num2

\*/

num2 = (num1 == 10) ? 100: 200;

System.out.println( "num2: "+num2);

/\* num1 is equal to 25 that's why

\* the first value is assigned

\* to the variable num2

\*/

num2 = (num1 == 25) ? 100: 200;

System.out.println( "num2: "+num2);

}

}

1. Instance of Operator

class operators

{

public static void main(String[] args)

{

Person obj1 = new Person();

Person obj2 = new Boy();

// As obj is of type person, it is not an

// instance of Boy or interface

System.out.println("obj1 instanceof Person: " +

(obj1 instanceof Person));

System.out.println("obj1 instanceof Boy: " +

(obj1 instanceof Boy));

System.out.println("obj1 instanceof MyInterface: " +

(obj1 instanceof MyInterface));

// Since obj2 is of type boy, whose parent class is

// person and it implements the interface Myinterface

// it is instance of all of these classes

System.out.println("obj2 instanceof Person: " +

(obj2 instanceof Person));

System.out.println("obj2 instanceof Boy: " +

(obj2 instanceof Boy));

System.out.println("obj2 instanceof MyInterface: " +

(obj2 instanceof MyInterface));

}

}

class Person

{

}

class Boy extends Person implements MyInterface

{

}

interface MyInterface

{

}

1. Precedence & associativity

public class operators

{

public static void main(String[] args)

{

int a = 20, b = 10, c = 0, d = 20, e = 40, f = 30;

// precedence rules for arithmetic operators.

// (\* = / = %) > (+ = -)

// prints a+(b/d)

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

// if same precendence then associative

// rules are followed.

// e/f -> b\*d -> a+(b\*d) -> a+(b\*d)-(e/f)

System.out.println("a+b\*d-e/f = "+(a + b \* d - e / f));

}

}

Java Program to Add two numbers

Java Program to Multiply two Numbers

Java Program to find Largest of three numbers using Ternary Operator

Java Program to find the smallest of three numbers using Ternary Operator

Java Program to check if number is positive or negative

Java Program to check whether number is even or odd