```
package com.itbulls.learnit.javacore.operators;

public class JavaOperators {

	public static void main(String[] args) {

		// ============ Arithmetic Operators

		// *** Unary Arithmetic Operators

//		+
//		-
//		++
//		--

		System.out.println("===== Unary Arithmetic Operators =====");

		int i = +10;
		int i2 = -10;
		int i3 = ++i;
		int i4 = i++;
		int i5 = --i;
		int i6 = i--;
		System.out.println("i = " + i);		// 10
		System.out.println("i2 = " + i2);  // -10
		System.out.println("i3 = " + i3);  // 11
		System.out.println("i4 = " + i4);  // 11
		System.out.println("i5 = " + i5);  // 11
		System.out.println("i6 = " + i6);  // 11
		System.out.println("i = " + i);	// 10
```

```
        System.out.println();

        // *** Binary Arithmetic Operators

//      +
//      -
//      /
//      %
//      *

        System.out.println("===== Binary Arithmetic Operators =====");

        int i7 = 10 + 1;
        int i8 = i7 - 1;
        int i9 = i8 / 2;
        int i10 = i9 % 2;
        int i11 = 5 % 10;
        int i12 = i10 * 3;

        System.out.println("i7 = " + i7);  // 11
        System.out.println("i8 = " + i8);  // 10
        System.out.println("i9 = " + i9);  // 5
        System.out.println("i10 = " + i10);          // 1
        System.out.println("i11 = " + i11);          // 5
        System.out.println("i12 = " + i12);          // 3
        System.out.println("Hello " + "World" + "!");

        System.out.println();
```

```
        // ============ Assignment Operators

//              =
//              +=
//              -=
//              *=
//              /=
//              %=

        System.out.println("===== Assignment Operators =====");

        int i13 = 10;
        i13 += 2;
        System.out.println(i13); // 12


        System.out.println();

        // ============ Relational Operators

//              ==
//              !=
//              >
//              <
//              >=
//              <=

        System.out.println("===== Relational Operators =====");

        int i14 = 10;
```

```
        int i15 = 20;

        System.out.println(i14 == i15);   // false
        System.out.println(i14 != i15);   // true
        System.out.println(i14 > i15);    // false
        System.out.println(i14 < i15);    // true
        System.out.println(i14 >= i15);   // false
        System.out.println(i14 <= i15);   // true

        System.out.println();

        // ============ Logical Operators

//      &
//      &&
//      |
//      ||
//      !
//      ^

        System.out.println("===== Logical Operators =====");

//      System.out.println(false & (5 / 0 == 0) ); // Runtime Exception
        System.out.println(false && (5 / 0 == 0) );         // false
//      System.out.println(true | (5 / 0 == 0) );   // Runtime Exception
        System.out.println(true || (5 / 0 == 0) ); // true
        System.out.println("!true = " + !true);             // false
        System.out.println(true ^ true);                    // false
        System.out.println(true ^ false);                   // true
        System.out.println(false ^ true);                   // true
```

```
        System.out.println();



        // ============ Bitwise Operators

//      &
//      |
//      ^
//      ~
//      >>
//      >>>
//      <<

        System.out.println("===== Bitwise Operators =====");

        System.out.println("4 & 5 = " + (4 & 5)); // 4
        /*
         *                      1 0 0
         *                      & & &
         *                      1 0 1
         *                      -----
         *                      1 0 0 = 4
         */

        System.out.println("4 | 5 = " + (4 | 5)); // 5
        /*
         *                      1 0 0
         *                      | | |
         *                      1 0 1
         *                      -----
         *                      1 0 1 = 5
```

```
 */

System.out.println("4 ^ 5 = " + (4 ^ 5)); // 1
/*
 *                                  1 0 0
 *                                  ^ ^ ^
 *                                  1 0 1
 *                                  -----
 *                                  0 0 1 = 1
 */

System.out.println("~1 = " + ~1);
System.out.println(Integer.toBinaryString(1));          // 00000001
System.out.println(Integer.toBinaryString(-2));         // 11111110
System.out.println((byte)0b11111110);                   // -2
System.out.println((byte)0b10000000);                   // -128 = -(2 ^ 7)
System.out.println((byte)0b11000000);                   // -64 = -(2 ^ 7) + (2
^ 6)


System.out.println("5 = " + 0b101);
System.out.println("5 >> 1 = " + (0b101 >> 1));
System.out.println("5 in binary format = " + Integer.toBinaryString(5));
System.out.println("2 in binary format = " + Integer.toBinaryString(2));    // 10


System.out.println("5 = " + 0b101);                                         //
000000101
System.out.println("5 >>> 1 = " + (0b101 >>> 1));          // 000000010

int negativeByteValue = 0b11111111111111111111111110000000;
```

```
		System.out.println("-128 = " + negativeByteValue);
	// 11111111111111111111111110000000
		System.out.println("-128 >>> 1 = " + (negativeByteValue >>> 1));//
01111111111111111111111111000000
		System.out.println("-128 >> 1 = " + (negativeByteValue >> 1));		//
11111111111111111111111111000000



		System.out.println("5 = " + 0b101);
		System.out.println("5 << 1 = " + (0b101 << 1));
		System.out.println("5 in binary format = " + Integer.toBinaryString(5));
		System.out.println("10 in binary format = " + Integer.toBinaryString(10));

		System.out.println("10 * 2 = 10 << 1 = " + (10 << 1));	// x << y = x * (2 ^ y)
		System.out.println("10 / 2 = 10 >> 1 = " + (10 >> 1));	// x >> y = x / (2 ^ y)

		System.out.println();

		// ============ Ternary Operator

//		(condition) ? true expression : false expression

		System.out.println("===== Ternary Operator =====");

		System.out.println(2 > 1 ? "2 is greater than one" : "2 is not less than one");
		System.out.println(2 < 1 ? "2 is greater than one" : "2 is not less than one");

		// ============ Operator Precedence
//
//		()
//		[]
//		.
```

```
//
//
//              ++
//              --
//
//
//              +
//              -
//              !
//              ~
//              ( type )
//
//
//              *
//              /
//              %
//
//
//              +
//              -
//
//
//              <<
//              >>
//              >>>
//
//
//              <
//              <=
//              >
//              >=
```

```
//                instanceof
//
//
//                ==
//                !=
//
//
//                &
//
//
//                ^
//
//
//                |
//
//
//                &&
//
//
//                ||
//
//
//                ? :
//
//
//                =
//                +=
//                -=
//                *=
//                /=
//                %=
```

```
        }

}
```