

JAVA FUNDAMENTALS COURSE

EXERCISE

OPERATORS IN JAVA

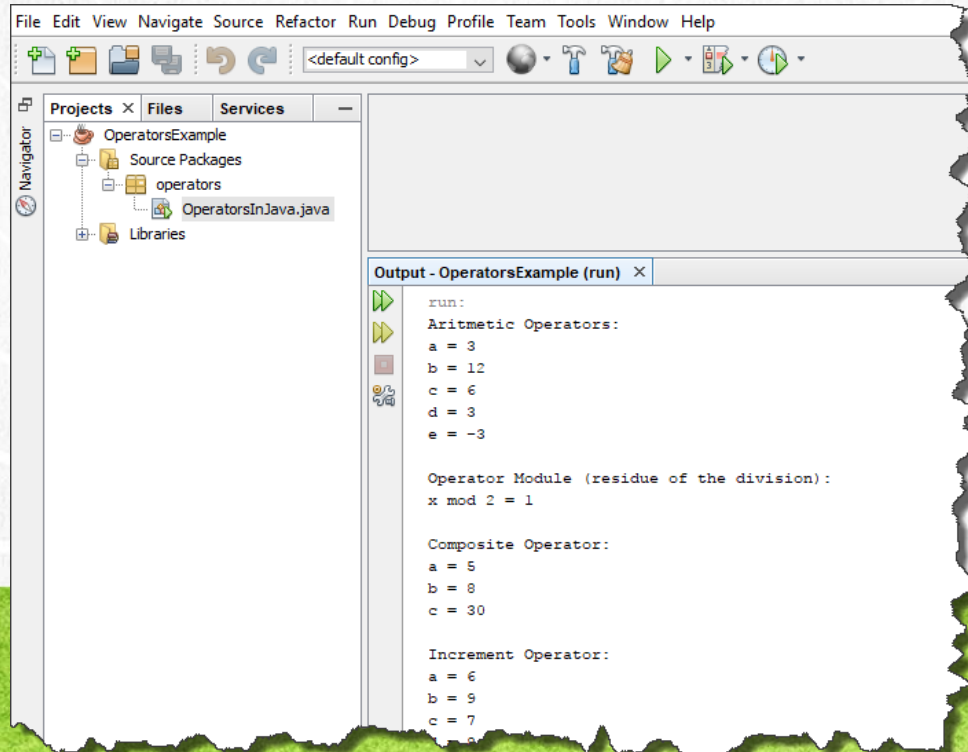


JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

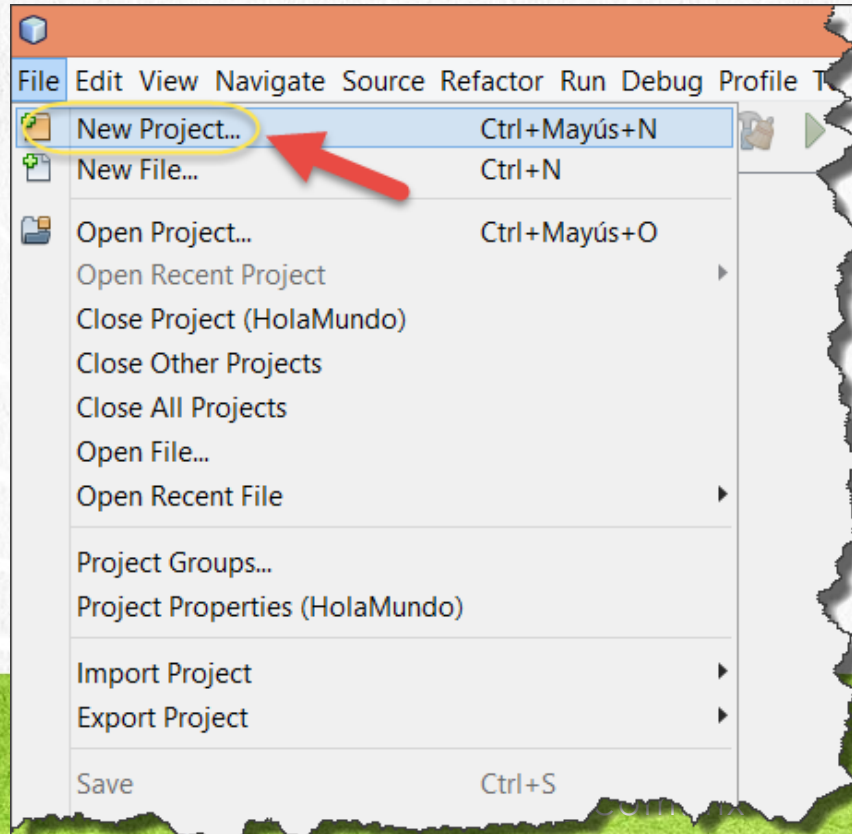
EXERCISE OBJECTIVE

Create a program to practice the Operators in Java. At the end we should observe the following:



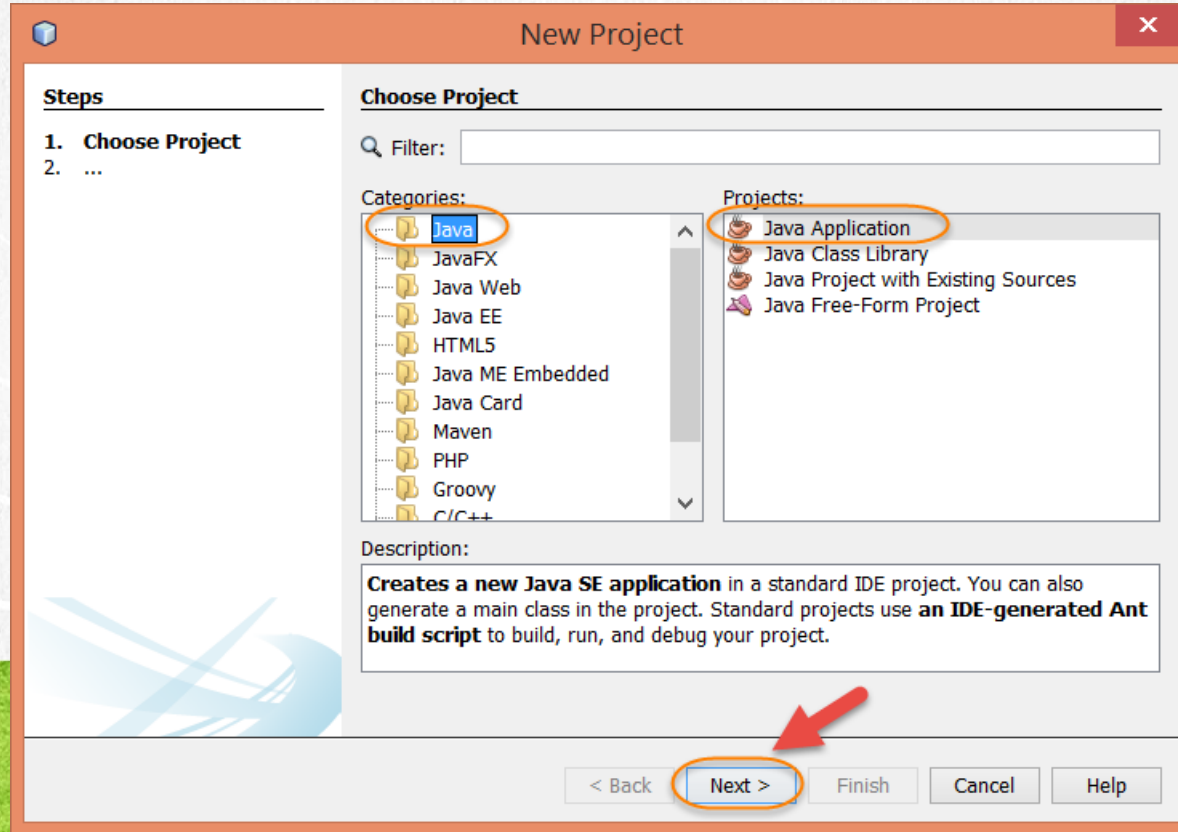
1. CREATE A NEW PROJECT

Create the OperatorsExample Project:



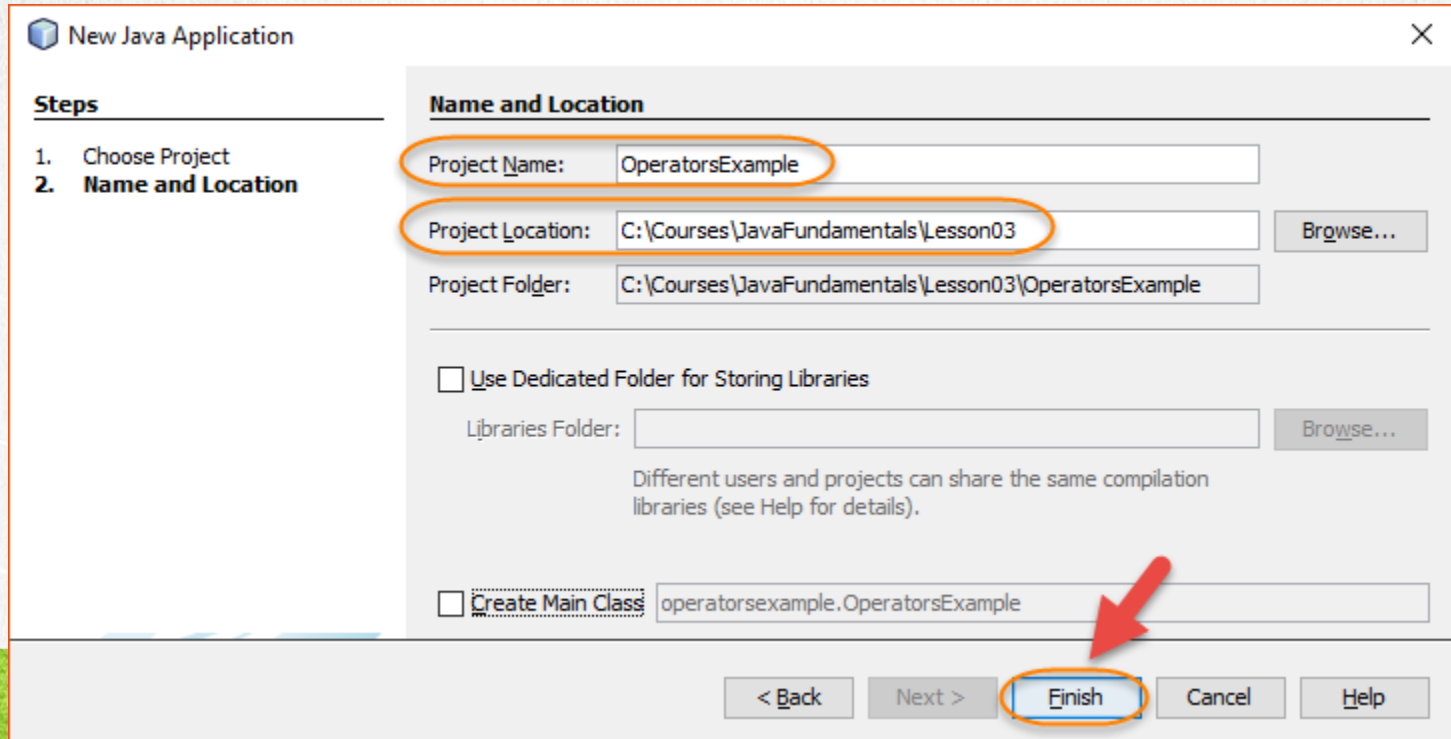
1. CREATE A NEW PROJECT (CONT)

Select Java -> Java Application and click on Next:



1. CREATE A NEW PROJECT (CONT)

Create the OperatorsExample Project:



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: OperatorsExample

Project Location: C:\Courses\JavaFundamentals\Lesson03 Browse...

Project Folder: C:\Courses\JavaFundamentals\Lesson03\OperatorsExample

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

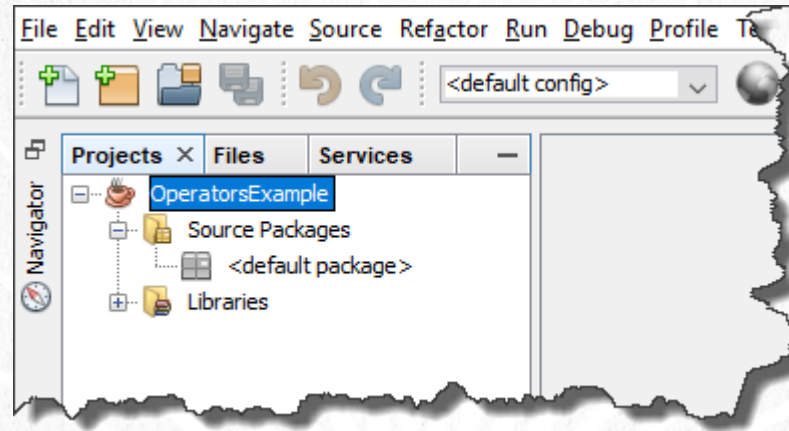
Different users and projects can share the same compilation libraries (see Help for details).

☐ Create Main Class operatorsexample.OperatorsExample

< Back Next > **Finish** Cancel Help

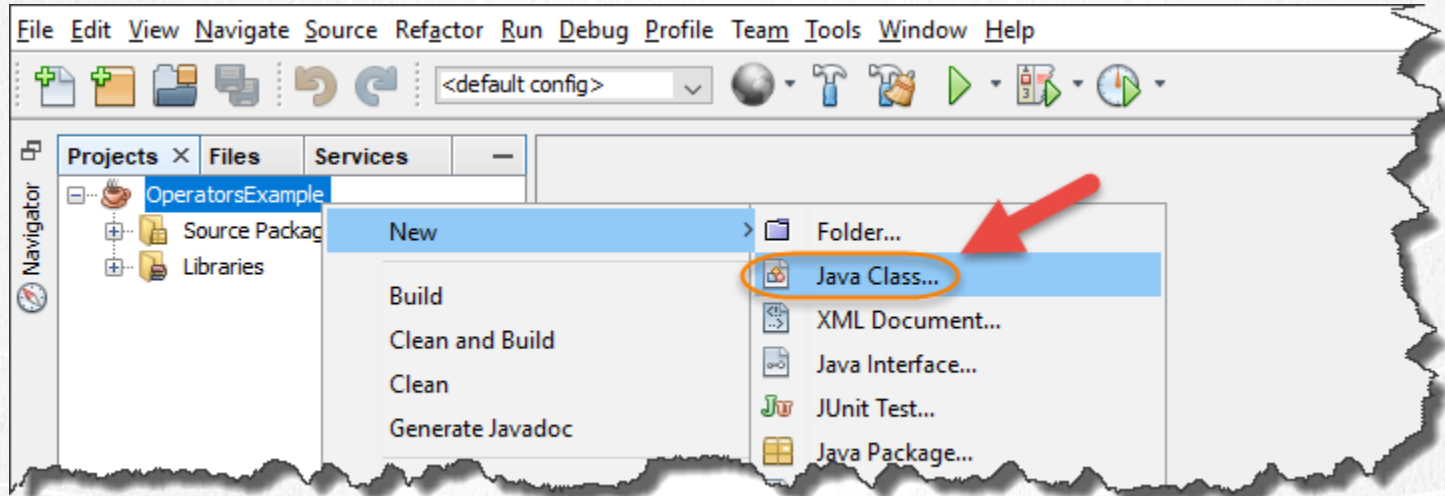
1. CREATE A NEW PROJECT (CONT)

The Project has the desired structure.



2. CREATE A CLASS

Create the Operators.java class. Right click on OperatorsExample Project -> New -> Java Class:



2. CREATE A CLASS (CONT)

Class Name: OperatorsInJava , package: operators and click on Finish.

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: OperatorsInJava

Project: OperatorsExample

Location: Source Packages

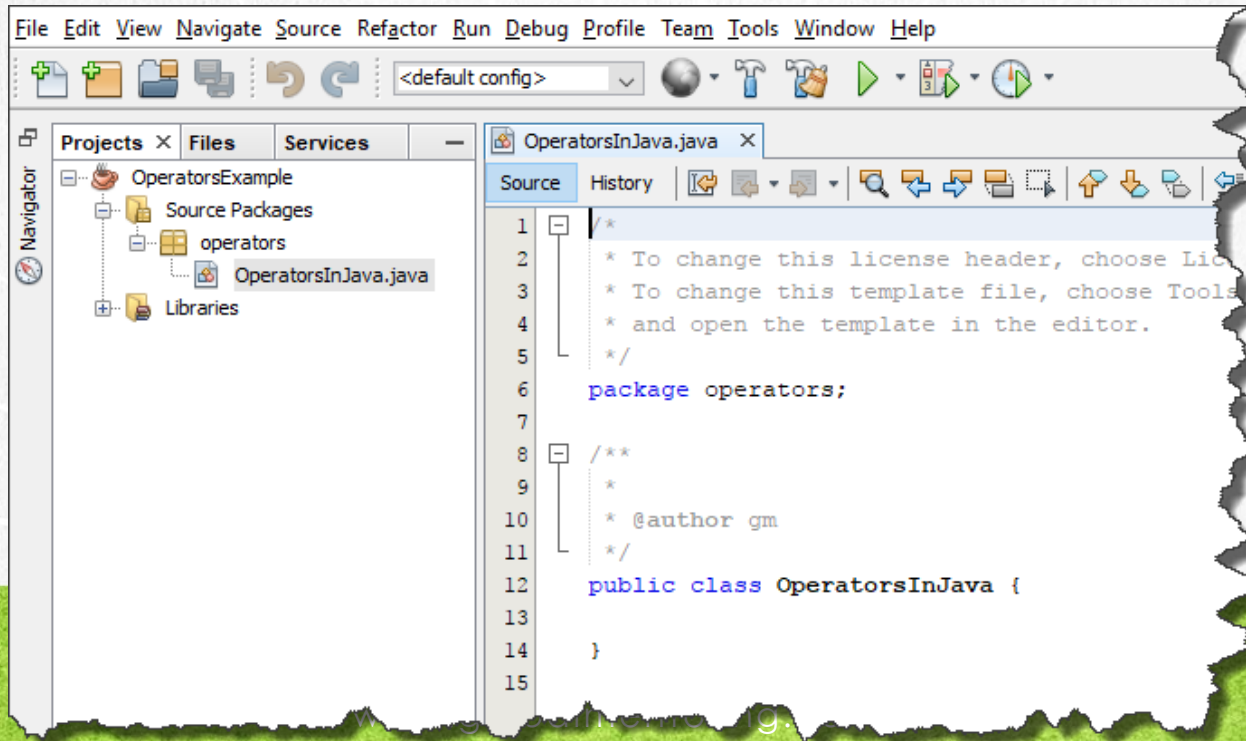
Package: operators

Created File: C:\Courses\JavaFundamentals\Lesson03\OperatorsExample\src\operators\OperatorsInJava.java

< Back Next > **Finish** Cancel Help

2. CREATE A CLASS (CONT)

This is the result of the created java class. We will substitute the code with the following:



PASO 3. MODIFY THE CODE

[Archivo OperatorsInJava.java:](#)

[Click to download the code](#)

```
package operators;

public class OperatorsInJava {

    public static void main(String[] args) {
        System.out.println("Aritmetic Operators:");
        int a = 1 + 2; //a=3
        int b = a * 4; //b=3*4 => b=12
        int c = b / 2; //c=12/2 => c=6 (the division gives an int result, not float)
        int d = c - a; //b=6-3 => d=3
        int e = -d; //e=-3
        System.out.println("a = " + a); //prints a=3
        System.out.println("b = " + b); //prints b=12
        System.out.println("c = " + c); //prints c=6
        System.out.println("d = " + d); //prints d=3
        System.out.println("e = " + e); //prints e=-3

        System.out.println("\nOperator Module (residue of the division):");
        System.out.println("x mod 2 = " + a % 2); //3%2 => 1 (int residue)

        System.out.println("\nComposite Operator:");
        a += 2; //a=a+2 => a=3+2 => a=5
        b -= 4; //b=b-4 => b=12-4 => b=8
        c *= a; //c=c*a => c=6*5 => c=30
        System.out.println("a = " + a); //prints a=5
        System.out.println("b = " + b); //prints b=8
        System.out.println("c = " + c); //prints c=30
    }
}
```

PASO 3. MODIFY THE CODE

Archivo OperatorsInJava.java:

[Click to download the code](#)

```
System.out.println("\nIncrement Operator:");
a++; //a=a+1
System.out.println("a = " + a); //prints a=6

//Preincrement/decrement (is increased before using the value)
c = ++a; //c=(a+1) => c=7
//Postincrement/decrement (is increased after using the value)
//The next time b is used, it is when it's increased
d = b++; //d=b => d=8
System.out.println("b = " + b); //prints b=9, b was waiting to increase the next time it was used
System.out.println("c = " + c); //prints c=7
System.out.println("d = " + d); //prints c=8

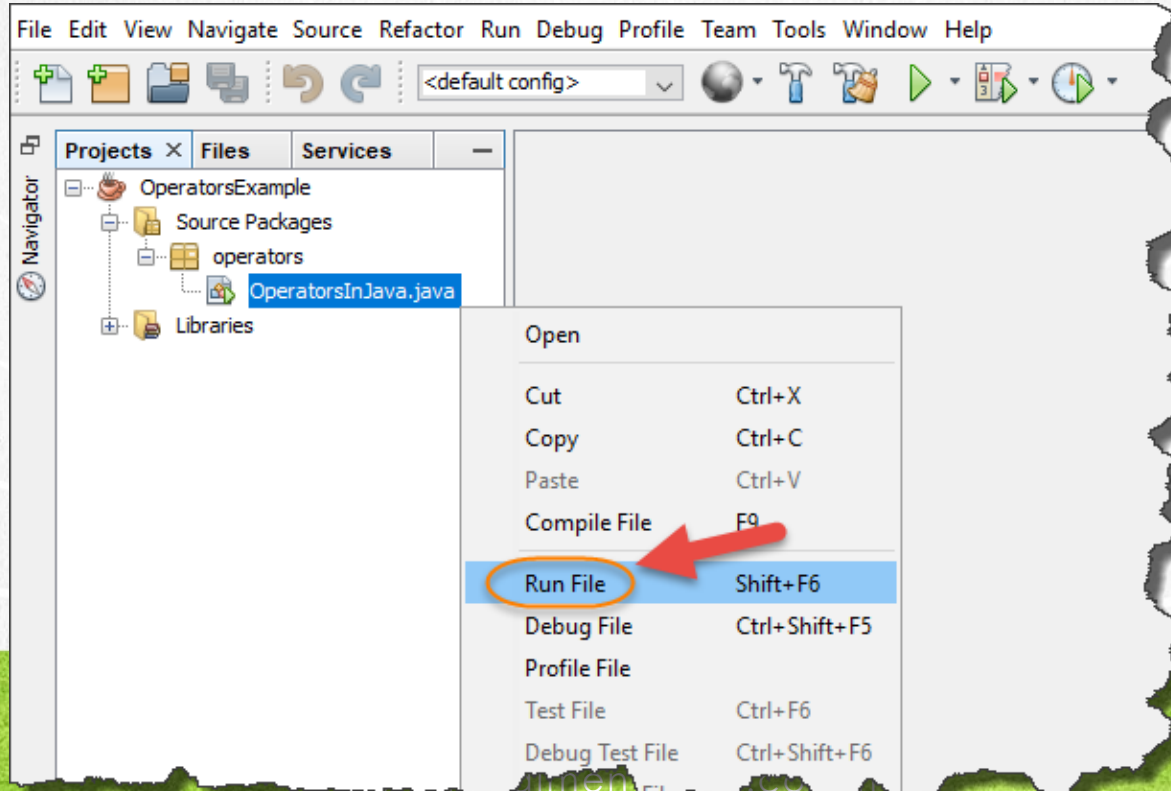
System.out.println("\nRelational operator:");
boolean res = a < b; //a is less than b => res=7<9 => res=true
System.out.println("res = " + res);

System.out.println("\nTernary Operator:");
int min = (a < b) ? a : b; //(7<9)=true => assign first value (a), if not assign second value (b)
System.out.println("min = " + min); //prints a => min = 7

System.out.println("\nAssignment Operator:");
int i, j, k;
//Many assignments at one time
i = j = k = 100; // i, j, and k are equal to 100
System.out.println("i = " + i); //prints i=100
System.out.println("j = " + j); //prints j=100
System.out.println("k = " + k); //prints k=100
}
```

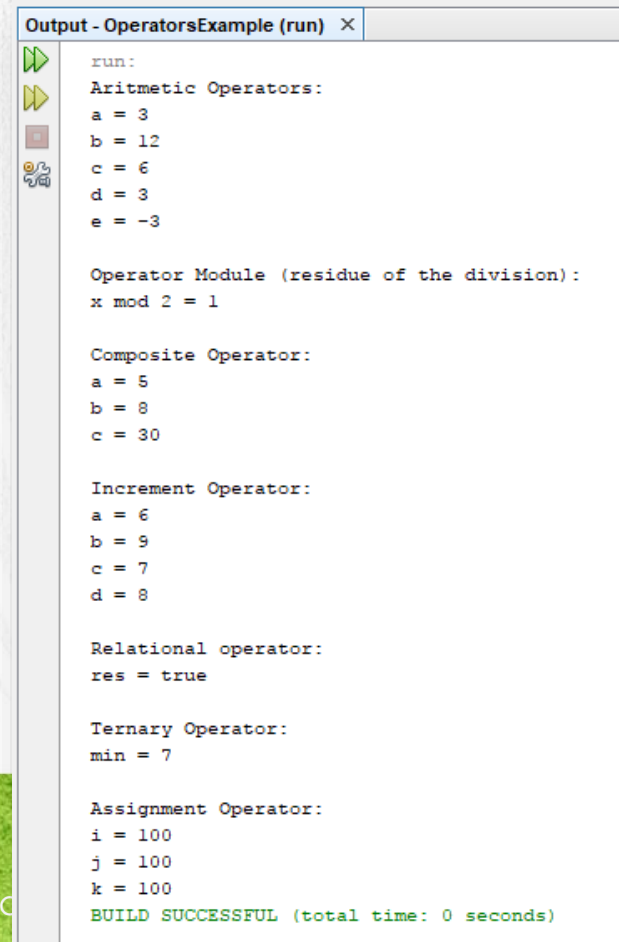

4. EXECUTE THE PROJECT

Execute our project. Right click over OperatorsInJava.java -> Run File:



4. EXECUTE THE PROJECT(CONT)

The result is as follows:



```
Output - OperatorsExample (run) X
run:
Arithmetic Operators:
a = 3
b = 12
c = 6
d = 3
e = -3

Operator Module (residue of the division):
x mod 2 = 1

Composite Operator:
a = 5
b = 8
c = 30

Increment Operator:
a = 6
b = 9
c = 7
d = 8

Relational operator:
res = true

Ternary Operator:
min = 7

Assignment Operator:
i = 100
j = 100
k = 100

BUILD SUCCESSFUL (total time: 0 seconds)
```

EXERCISE CONCLUSION

- With this exercise we have put into practice some of the most common operators in Java.
- For more information about the operators in Java, consult:
- <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>
- In the following exercise we will see the use of operator precedence.

ONLINE COURSE

JAVA FUNDAMENTALS

Author: Eng. Ubaldo Acosta



JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx