

JAVA FUNDAMENTALS COURSE

EXERCISE

ARITHMETIC PROJECT V3

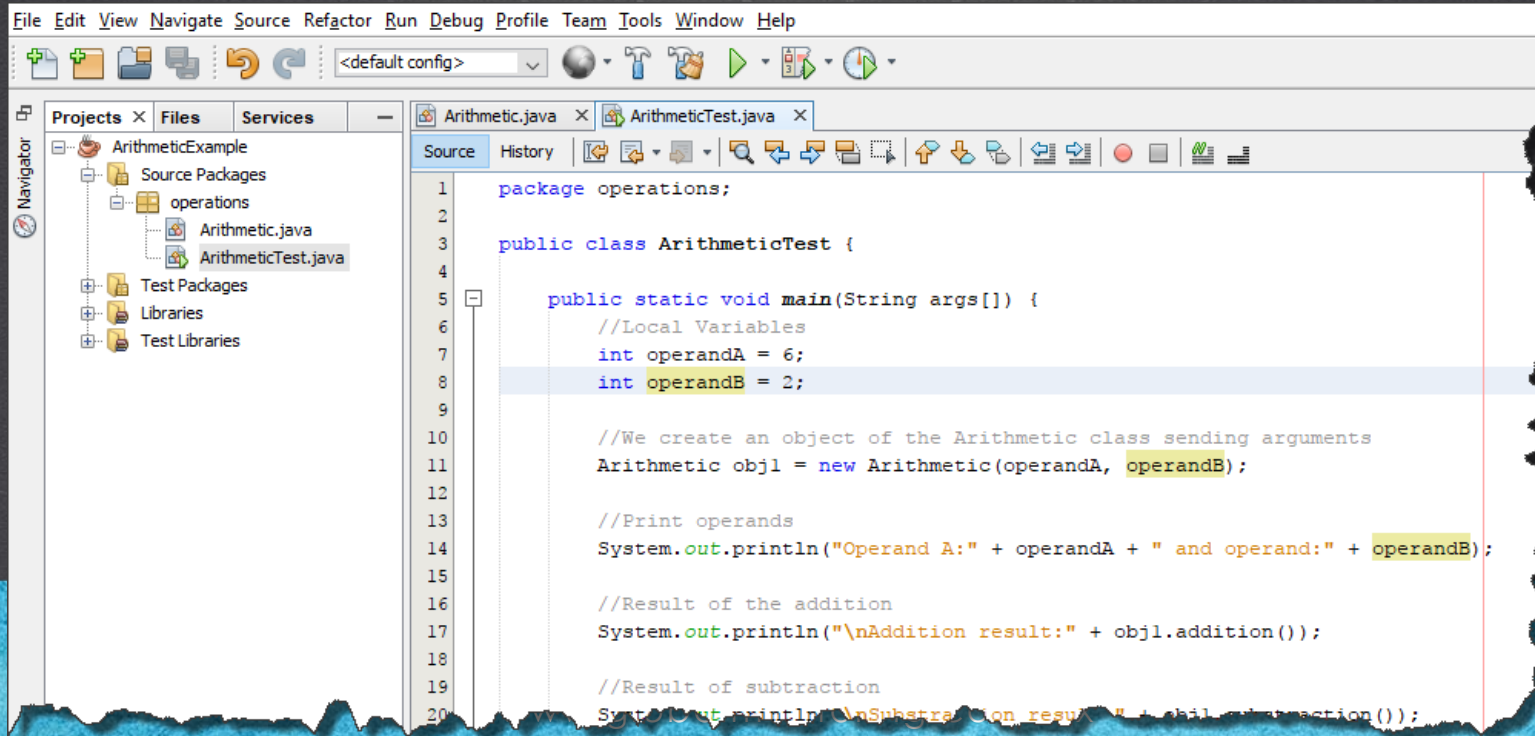


JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Create the Arithmetic exercise. At the end we should observe the following:

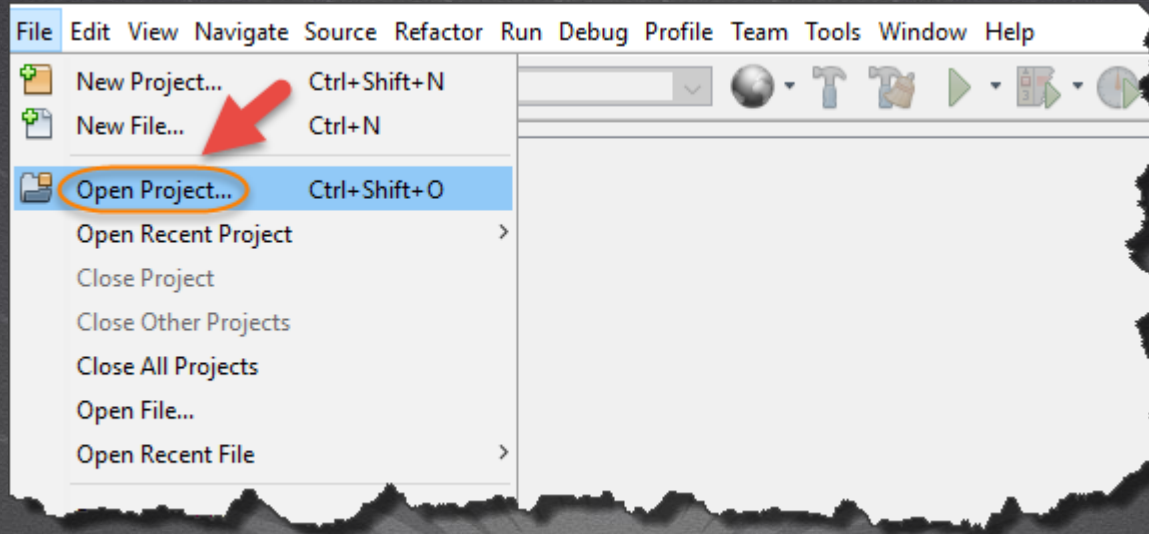


The screenshot shows an IDE window with the following structure:

- Navigator:**
 - ArithmeticExample
 - Source Packages
 - operations
 - Arithmetic.java
 - ArithmeticTest.java
 - Test Packages
 - Libraries
 - Test Libraries
- Editor:** ArithmeticTest.java
 - Line 1: `package operations;`
 - Line 2: (empty)
 - Line 3: `public class ArithmeticTest {`
 - Line 4: (empty)
 - Line 5: `public static void main(String args[]) {`
 - Line 6: `//Local Variables`
 - Line 7: `int operandA = 6;`
 - Line 8: `int operandB = 2;`
 - Line 9: (empty)
 - Line 10: `//We create an object of the Arithmetic class sending arguments`
 - Line 11: `Arithmetic obj1 = new Arithmetic(operandA, operandB);`
 - Line 12: (empty)
 - Line 13: `//Print operands`
 - Line 14: `System.out.println("Operand A:" + operandA + " and operand:" + operandB);`
 - Line 15: (empty)
 - Line 16: `//Result of the addition`
 - Line 17: `System.out.println("\nAddition result:" + obj1.addition());`
 - Line 18: (empty)
 - Line 19: `//Result of subtraction`
 - Line 20: `System.out.println("\nSubtraction result:" + obj1.subtraction());`

1. OPEN THE PROJECT

Open the Arithmetic project if necessary:

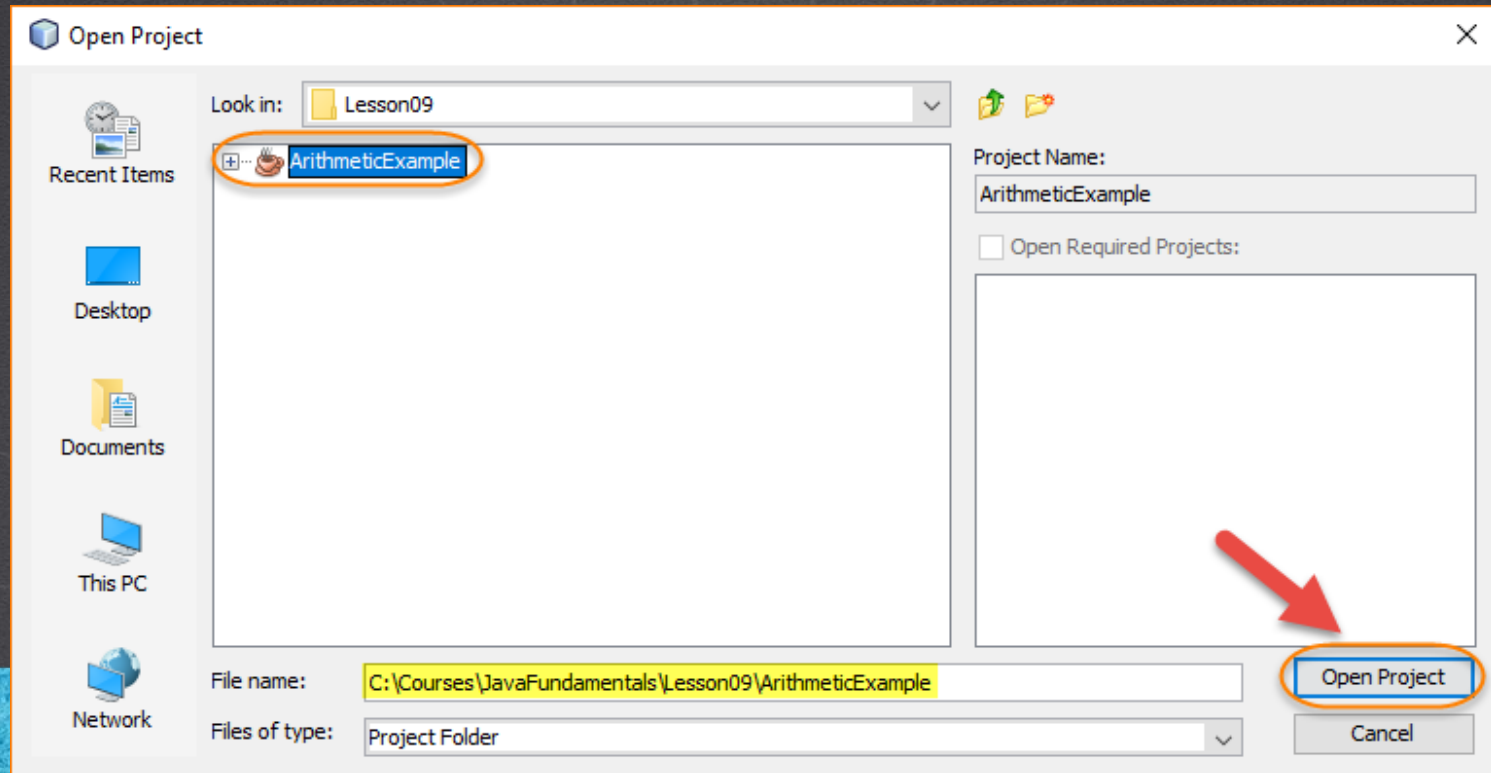


JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

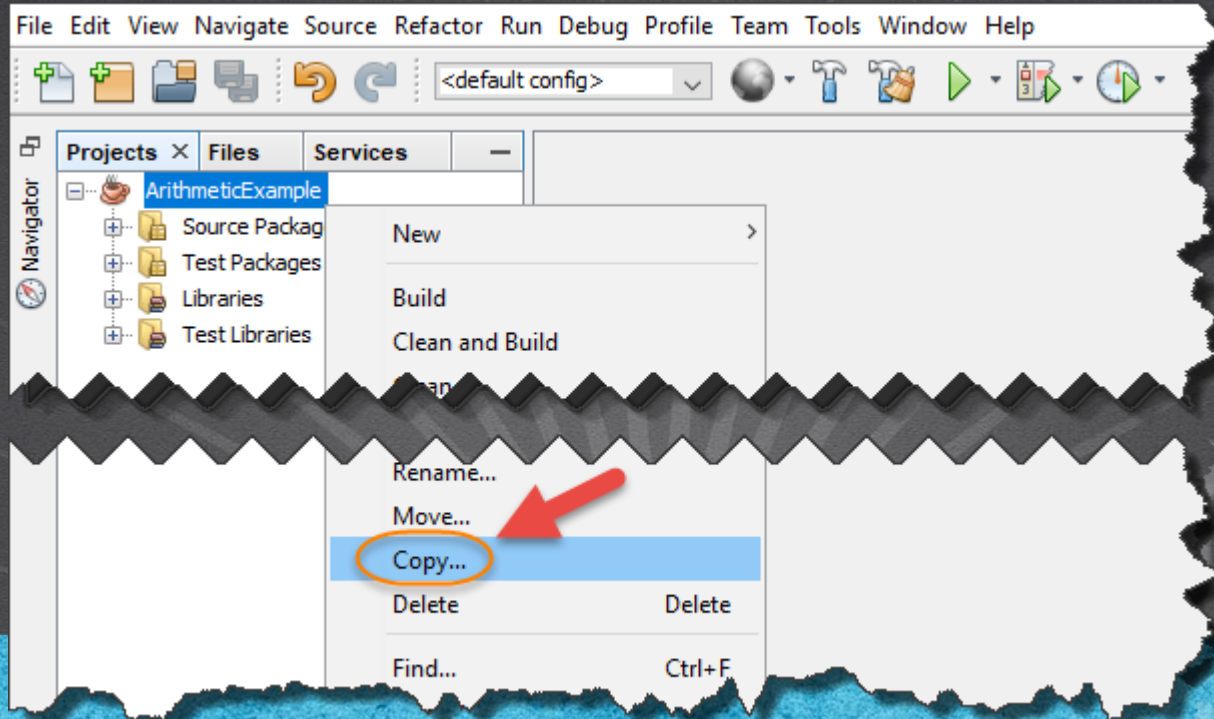
1. OPEN THE PROJECT

Open the Arithmetic project if necessary:



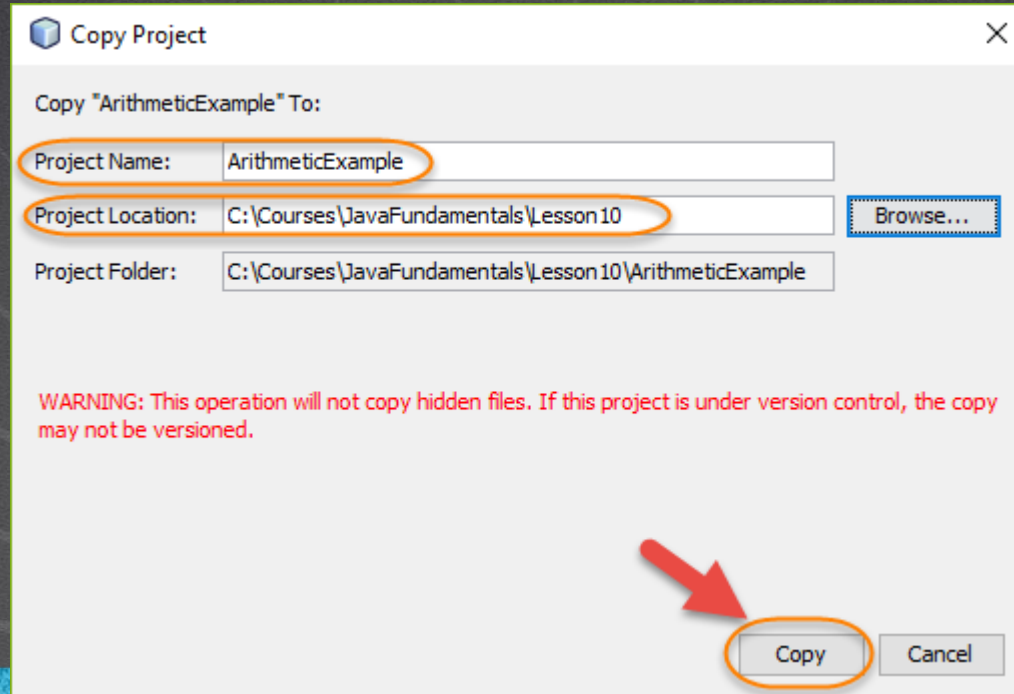
2. PROJECT CLONING

We are going to copy the Arithmetic project:



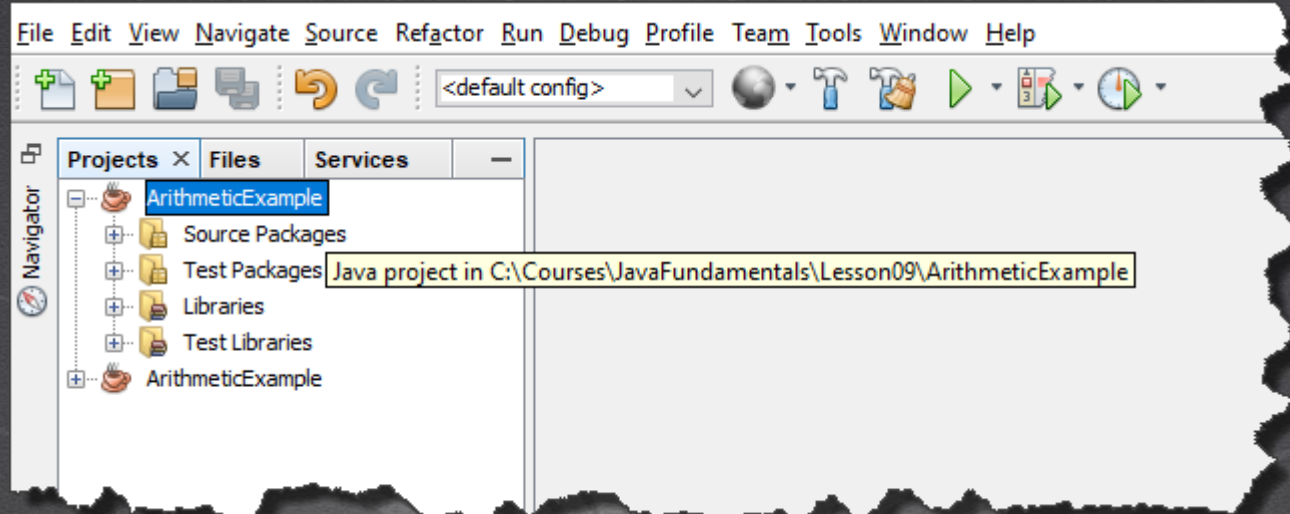
2. PROJECT CLONING

We are going to copy the Arithmetic project:



3. CLOSE PROJECT

We are going to close the Arithmetic project of the previous lesson:

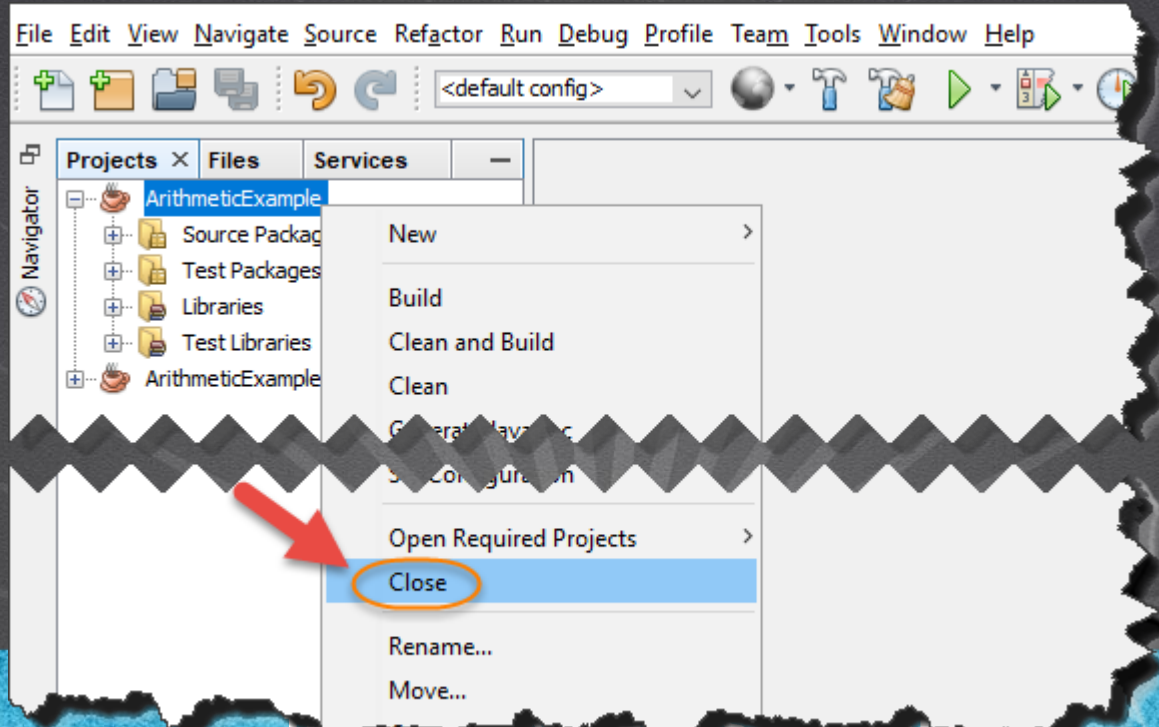


JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

3. CLOSE PROJECT

We are going to close the Arithmetic project of the previous lesson:



4. MODIFY THE CODE

Modify the Arithmetic class to add more methods:

Arithmetic.java:

```
package operations;

public class Arithmetic {

    //Attributes of the class
    int a;
    int b;

    //Empty Constructor
    //Remember that if we add a different constructor to the empty one
    //this constructor is not created anymore and we must create it if we need it
    Arithmetic() {
    }

    //Constructor with 2 arguments
    Arithmetic(int a, int b) {
        //Use of this operator
        this.a = a;
        this.b = b;
    }
}
```

4. MODIFY THE CODE

Modify the Arithmetic class to add more methods:

Arithmetic.java:

```
//This method takes the attributes of the class to do the addition
int addition() {
    return a + b;
}

//Substraction method
int subtraction() {
    return a - b;
}

//Multiplication method
int multiplication() {
    return a * b;
}

//Division method
int division() {
    return a / b;
}
}
```

5. MODIFY THE CODE

Modify the ArithmeticTest.java:

```
package operations;

public class ArithmeticTest {

    public static void main(String args[]) {
        //Local Variables
        int operandA = 6;
        int operandB = 2;

        //We create an object of the Arithmetic class sending arguments
        Arithmetic obj1 = new Arithmetic(operandA, operandB);

        //Print operands
        System.out.println("Operand A:" + operandA + " and operand B:" + operandB);

        //Result of the addition
        System.out.println("\nAddition result:" + obj1.addition());

        //Result of subtraction
        System.out.println("\nSubtraction result:" + obj1.subtraction());
    }
}
```


5. MODIFY THE CODE

Modify the ArithmeticTest.java:

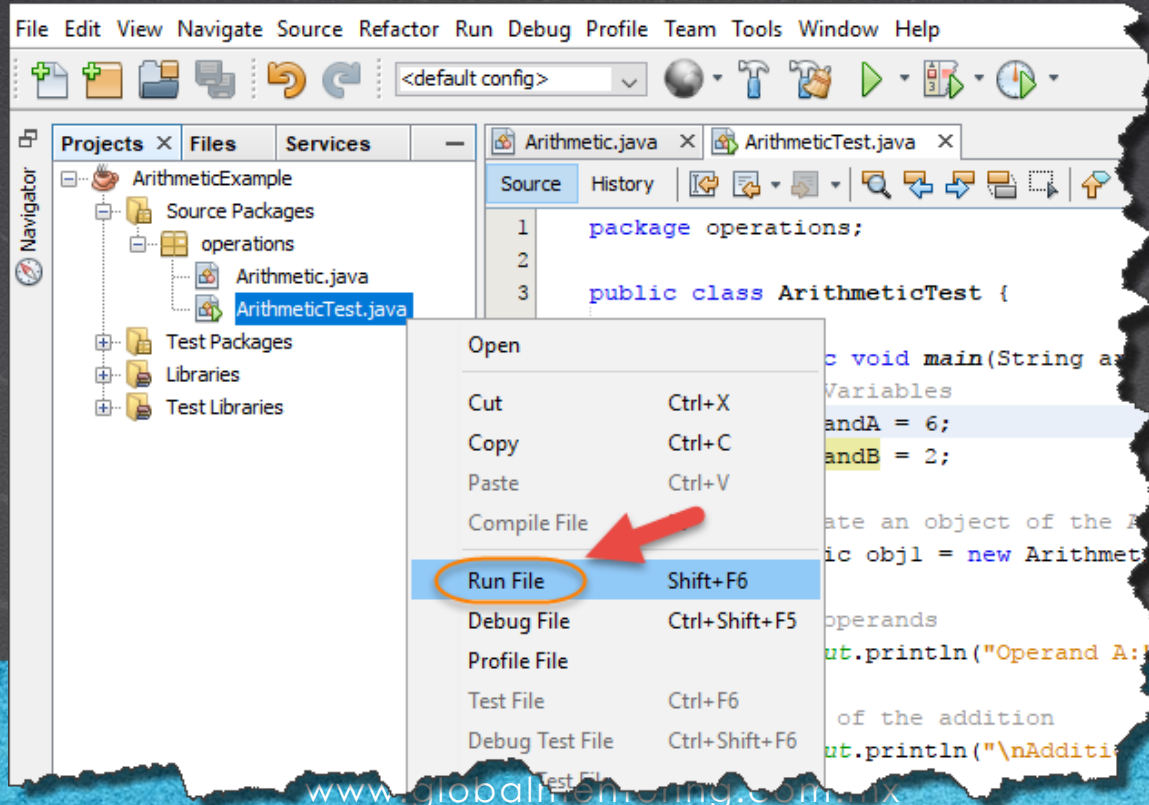
```
//Result of multiplication
System.out.println("\nMultiplication result:" + obj1.multiplication());

//Result of division
System.out.println("\nDivision result:" + obj1.division());
}

// void anotherMethod(){
//     //Cannot access a local variable declared in another method
//     System.out.println("Operand A" + operandA);
// }
}
```

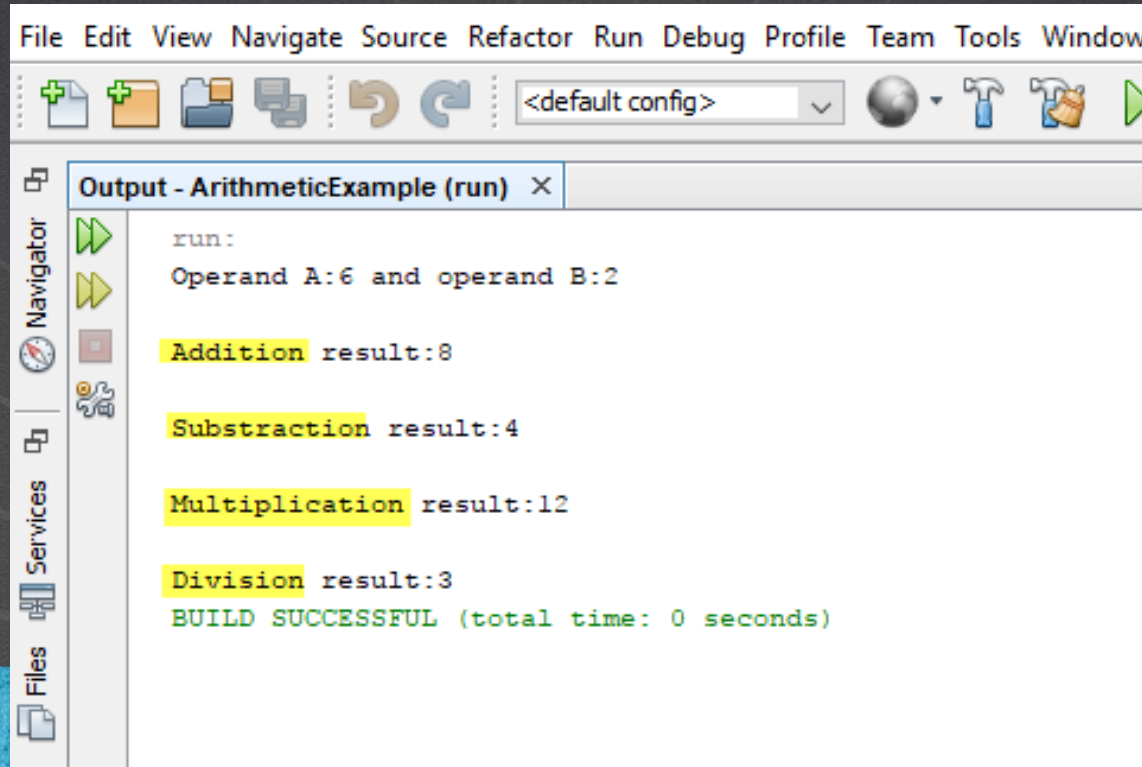
6. EXECUTE THE PROJECT

We execute our project. We give right click -> Run:



6. EXECUTE THE PROJECT

The result is as follows:



The screenshot shows an IDE window titled "Output - ArithmeticExample (run)". The output text is as follows:

```
run:  
Operand A:6 and operand B:2  
  
Addition result:8  
  
Substraction result:4  
  
Multiplication result:12  
  
Division result:3  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window) and a toolbar with icons for file operations, running, and debugging. On the left, there is a sidebar with icons for Navigator, Services, and Files.

EXERCISE CONCLUSION

- With this exercise we have put into practice some topics such as the creation of methods, the use of this operator, as well as the issue of scope of variables.
- With these concepts we have added some improvements to our Arithmetic class, as well as some methods that complement the functionality of the class.
- In later topics we will continue working on more concepts about this same project.

ONLINE COURSE

JAVA FUNDAMENTALS

By: Eng. Ubaldo Acosta



JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx