JAVA FUNDAMENTALS COURSE

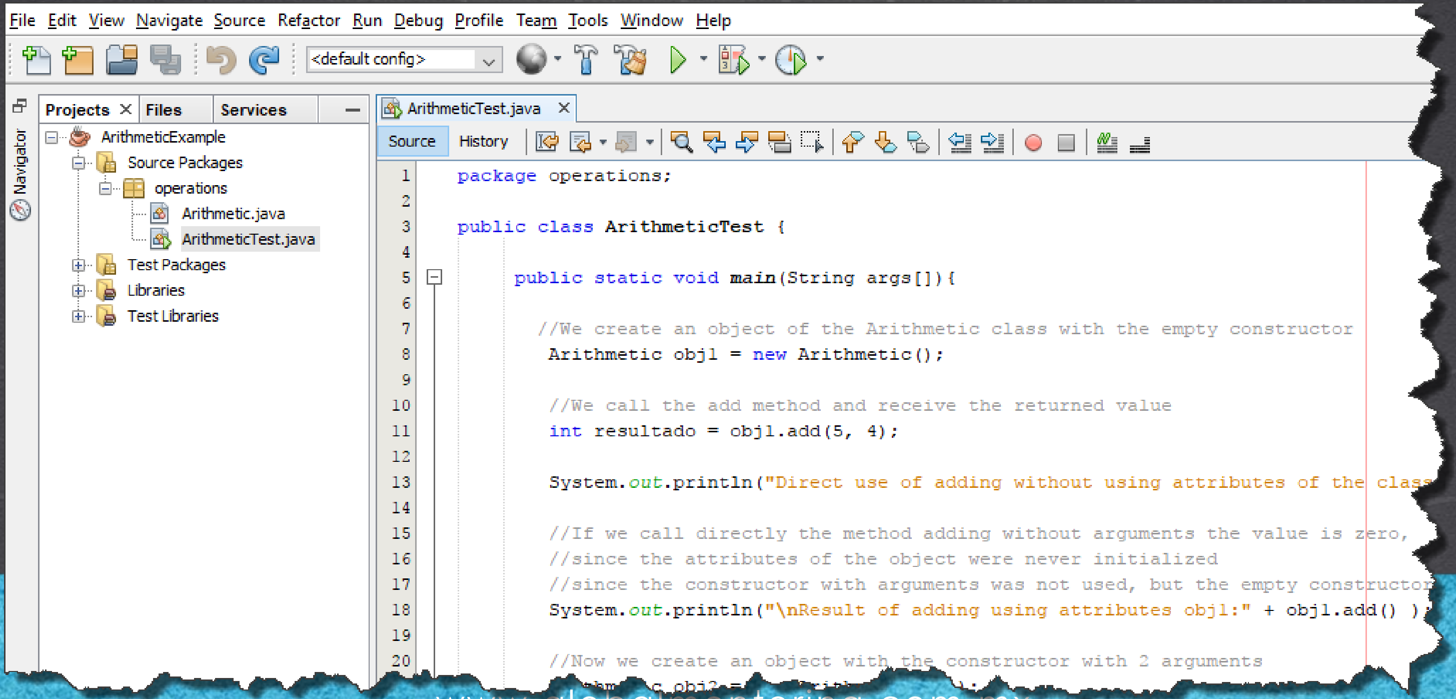
EXERCISE

ARITHMETIC PROJECT WITH CONSTRUCTORS

Global Mentoring

EXERCISE OBJECTIVE

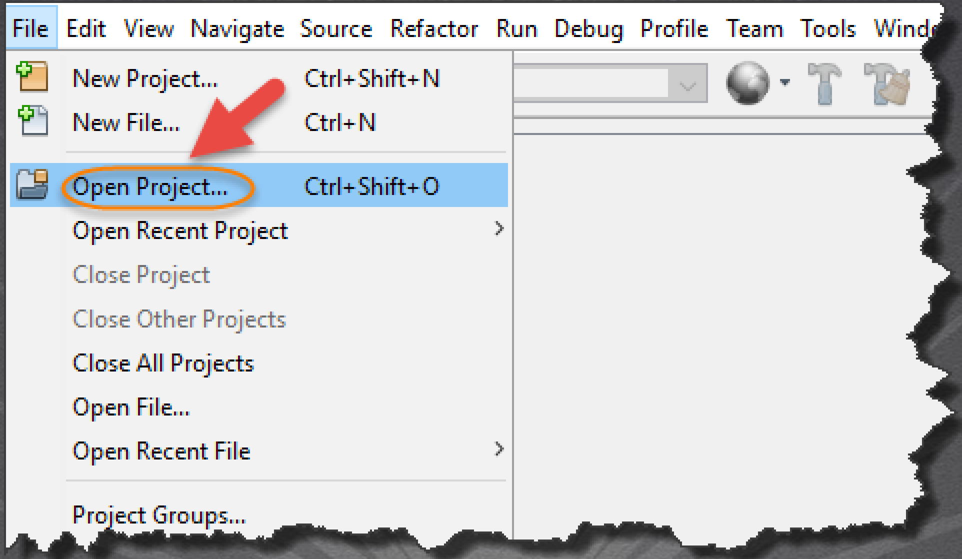
Create the Arithmetic exercise with constructors. At the end we should observe the following:



```java
package operations;

public class ArithmeticTest {

    public static void main(String args[]){

      //We create an object of the Arithmetic class with the empty constructor
       Arithmetic objl = new Arithmetic();

       //We call the add method and receive the returned value
       int resultado = objl.add(5, 4);

       System.out.println("Direct use of adding without using attributes of the clas

       //If we call directly the method adding without arguments the value is zero,
       //since the attributes of the object were never initialized
       //since the constructor with arguments was not used, but the empty constructor
       System.out.println("\nResult of adding using attributes objl:" + objl.add() );

       //Now we create an object with the constructor with 2 arguments
         thm   c obj2 =    rith        );
```

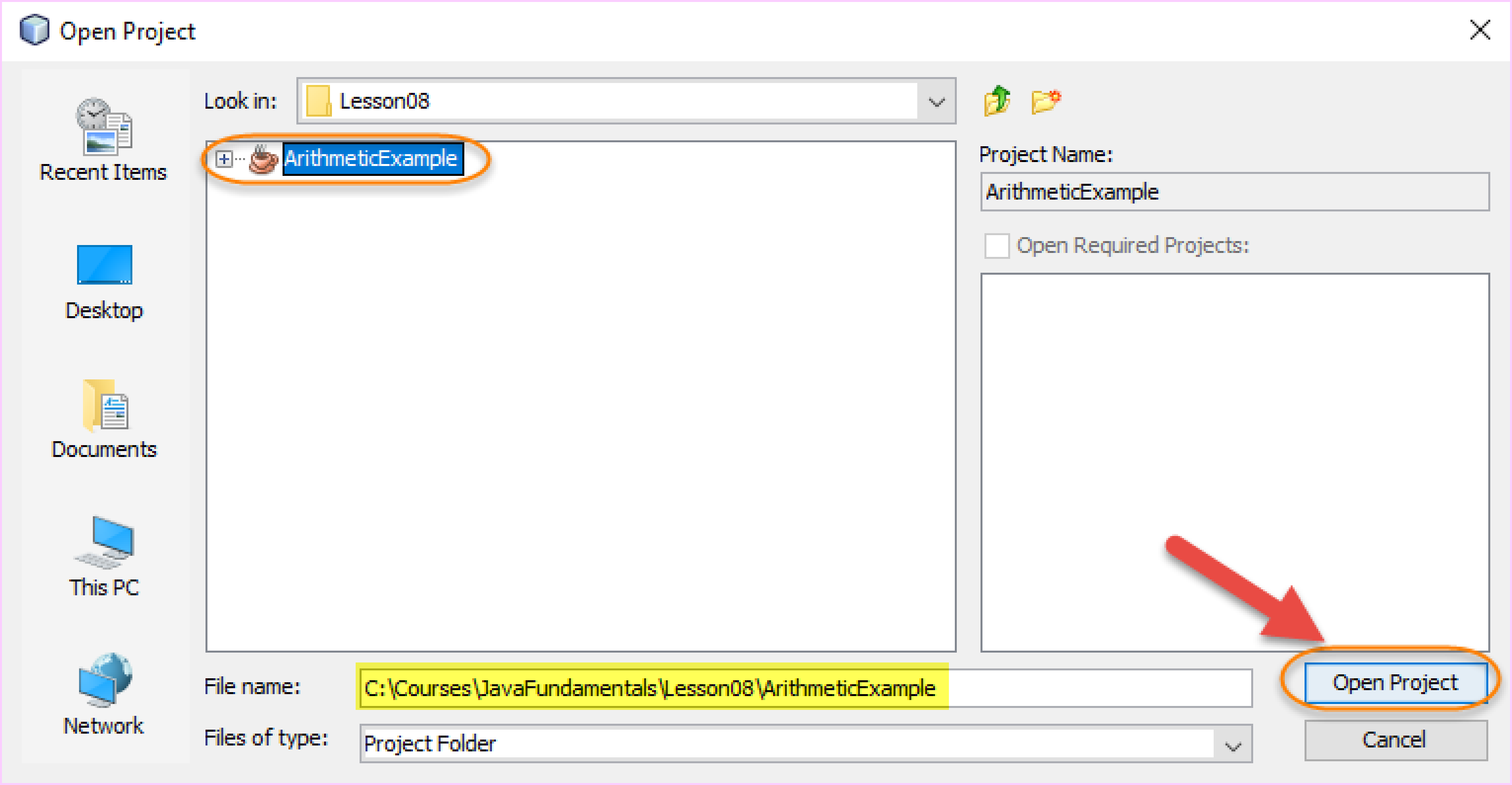
1. OPEN PROJECT

If necessary open the Arithmetic project:



1. OPEN PROJECT

If necessary open the Arithmetic project:



Open Project

Look in: Lesson08

ArithmeticExample

Project Name: ArithmeticExample

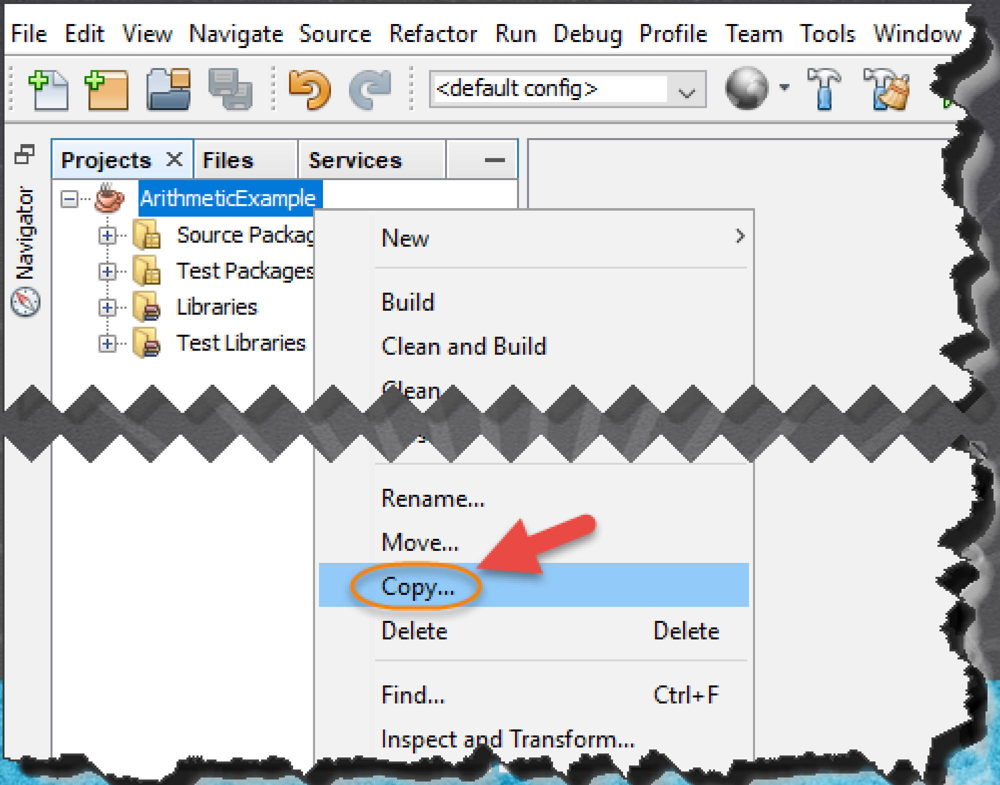
□ Open Required Projects:

File name: C:\Courses\JavaFundamentals\Lesson08\ArithmeticExample

Files of type: Project Folder

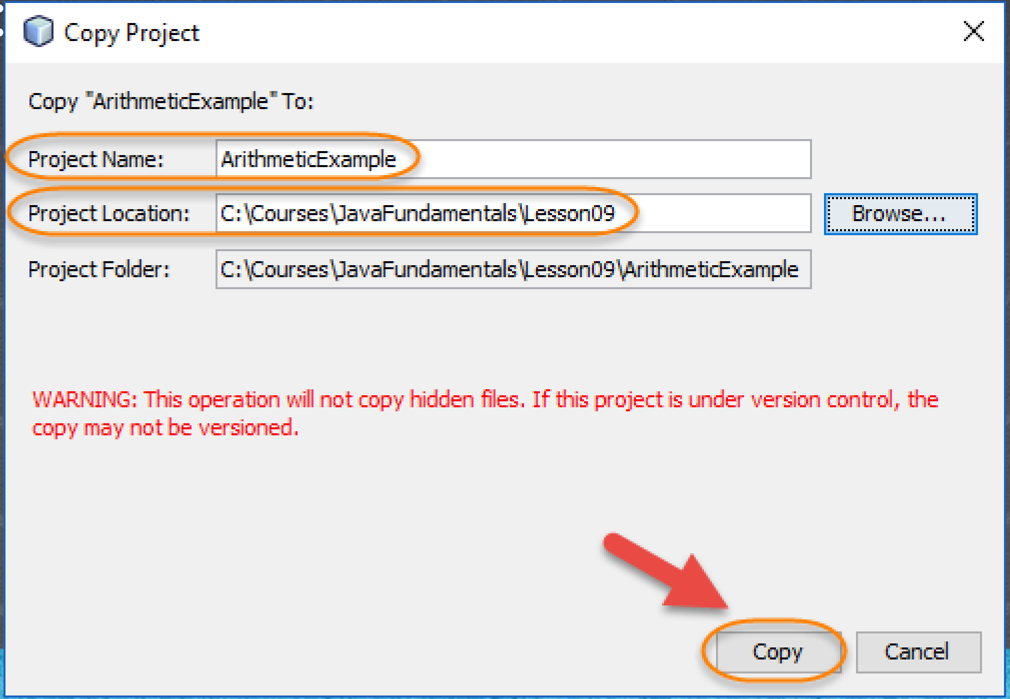
2. COPY PROJECT

Copy the project to a new location:



2. COPY PROJECT

The Project name is the same, but change the location as shown below:



Copy Project

Copy "ArithmeticExample" To:

Project Name: ArithmeticExample

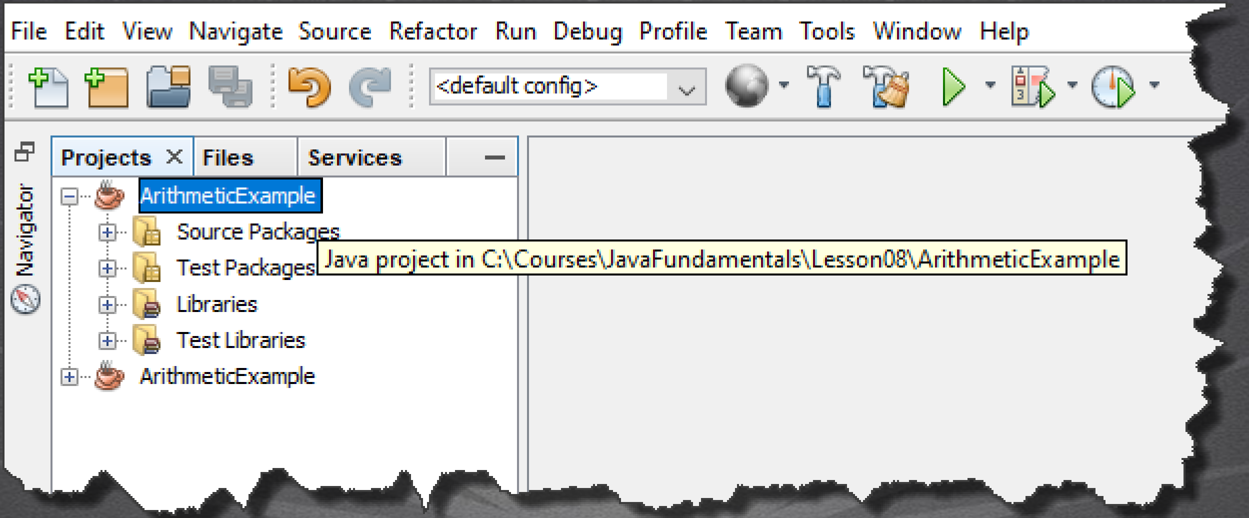
Project Location: C:\Courses\JavaFundamentals\Lesson09

Project Folder: C:\Courses\JavaFundamentals\Lesson09\ArithmeticExample

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

3. CLOSE THE PROJECT

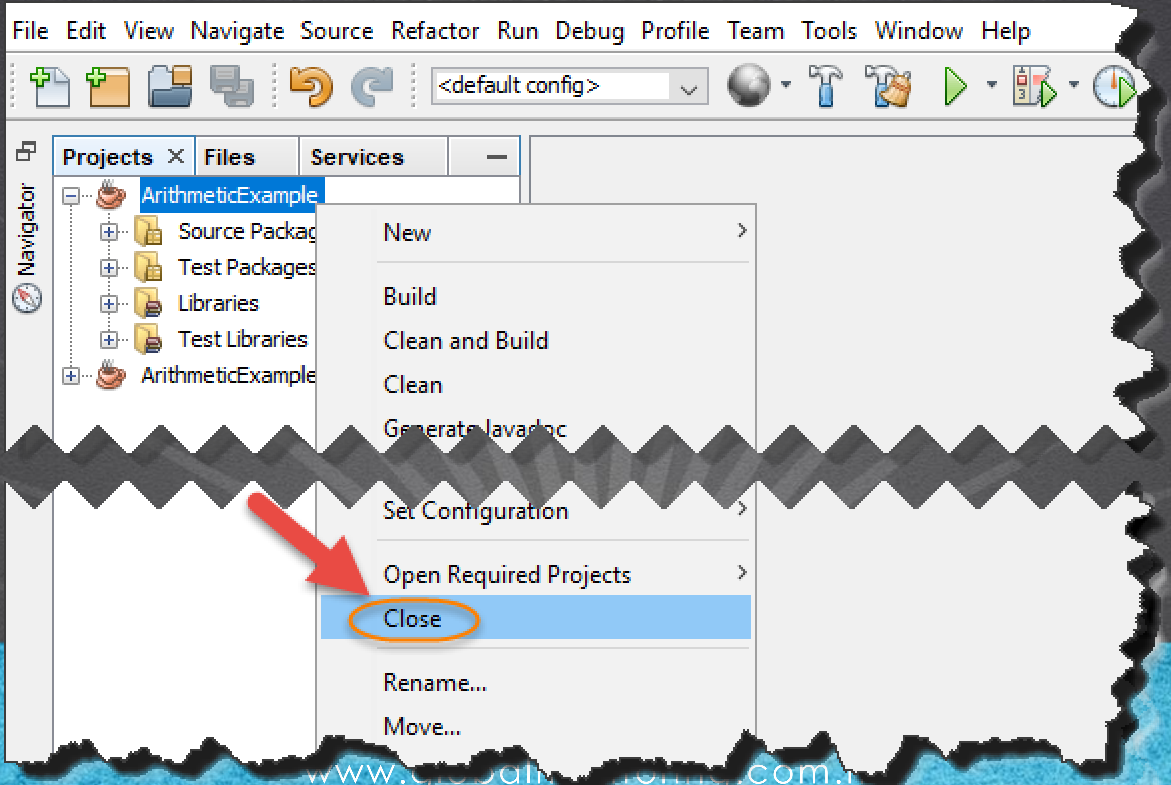
Close the previous Project. First we identify the project to be closed, leaving the cursor on the project to identify it:



Java project in C:\Courses\JavaFundamentals\Lesson08\ArithmeticExample

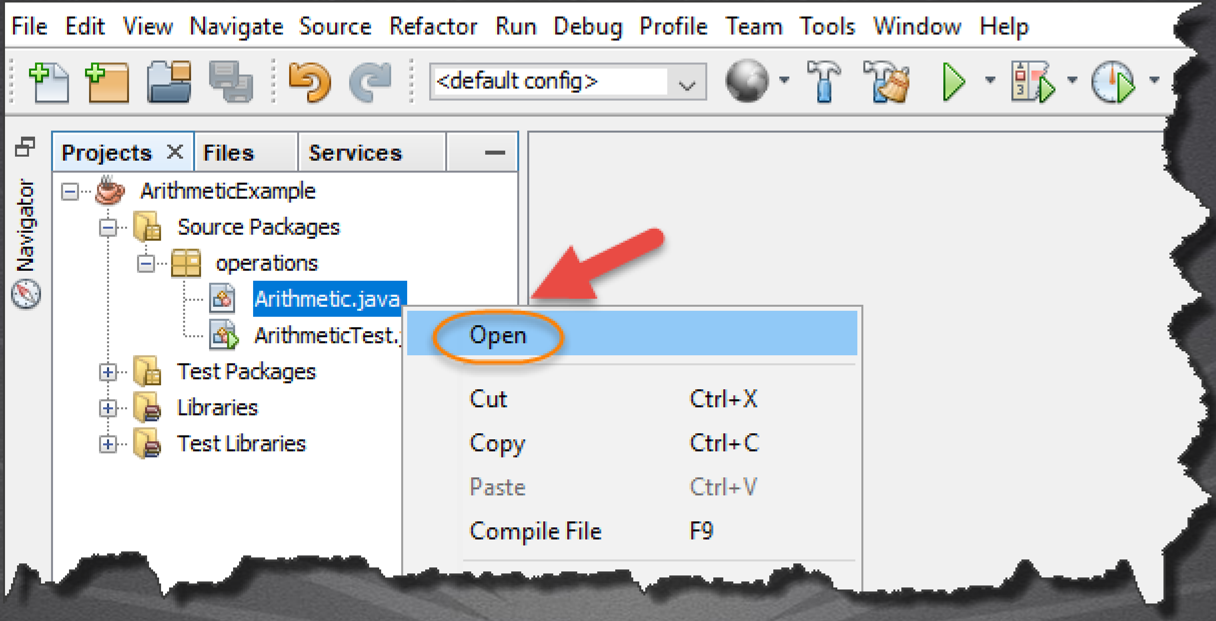
3. CLOSE THE PROJECT

We close the previous project and leave open the new one:



4. MODIFY A CLASS

Modify the Arithmetic class to add the Constructors as follow:



4. MODIFY THE CODE

Arithmetic.java:

```java
package operations;

public class Arithmetic {

    //Class attributes
    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 arg1, int arg2) {
        a = arg1;
        b = arg2;
    }
```

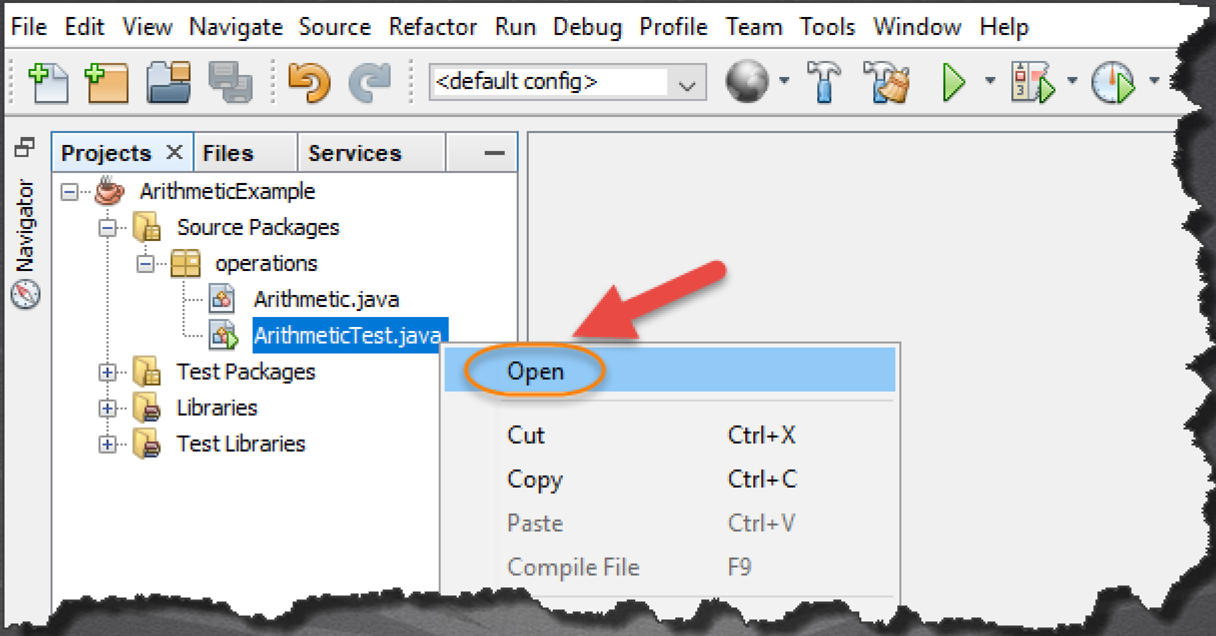
4. MODIFY THE CODE (CONT)

Arithmetic.java:

```java
    //This method takes new values, not the attributes of the class
    int add(int a, int b) {
        return a + b;
    }

    //This method takes the attributes of the class to make the addition.
    //Adding more than one method with the same name is known as overloading
    //of methods, we will study later this topic.
    int add() {
        return a + b;
    }
}
```

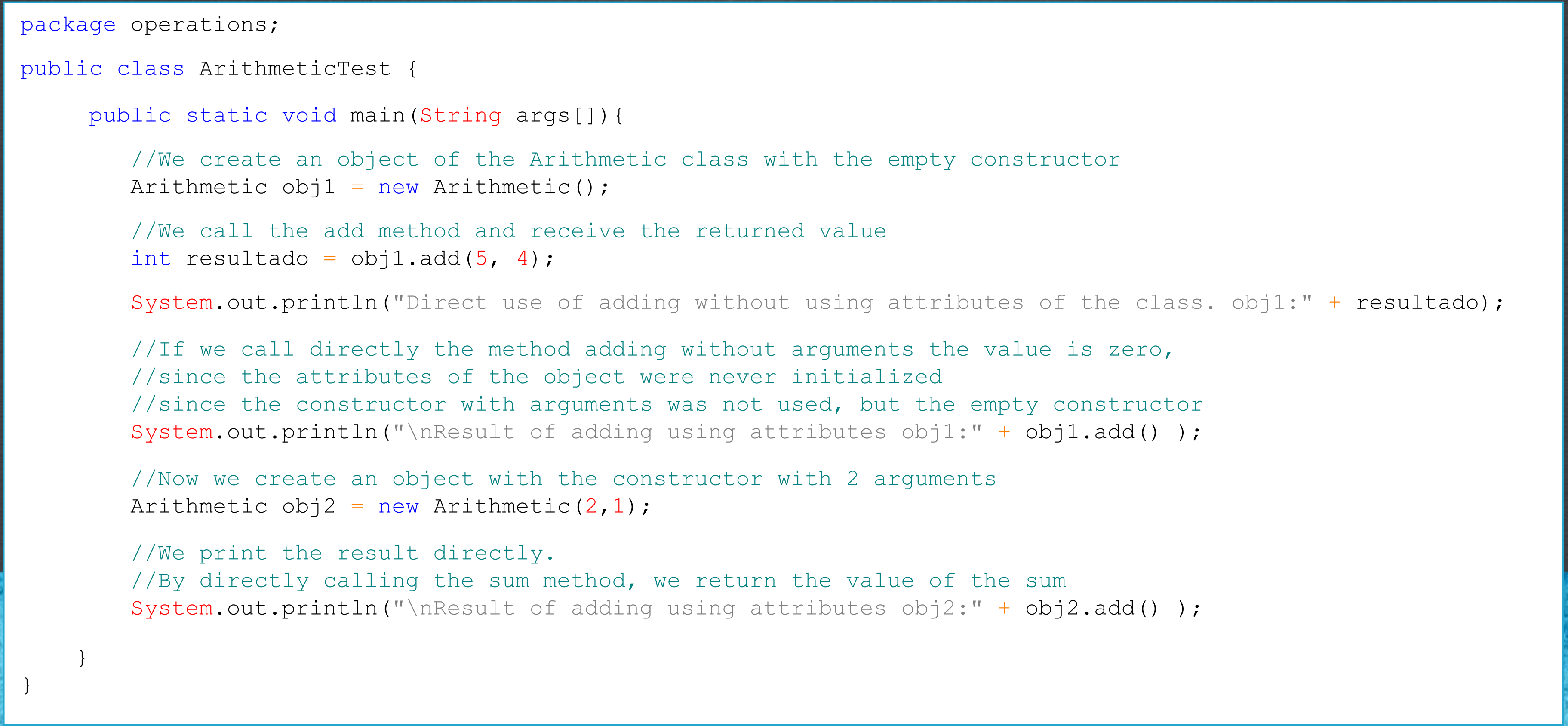
5. MODIFY A CLASS

Modify the ArithmeticTest class to add more objects as follows:



5. MODIFY THE CODE

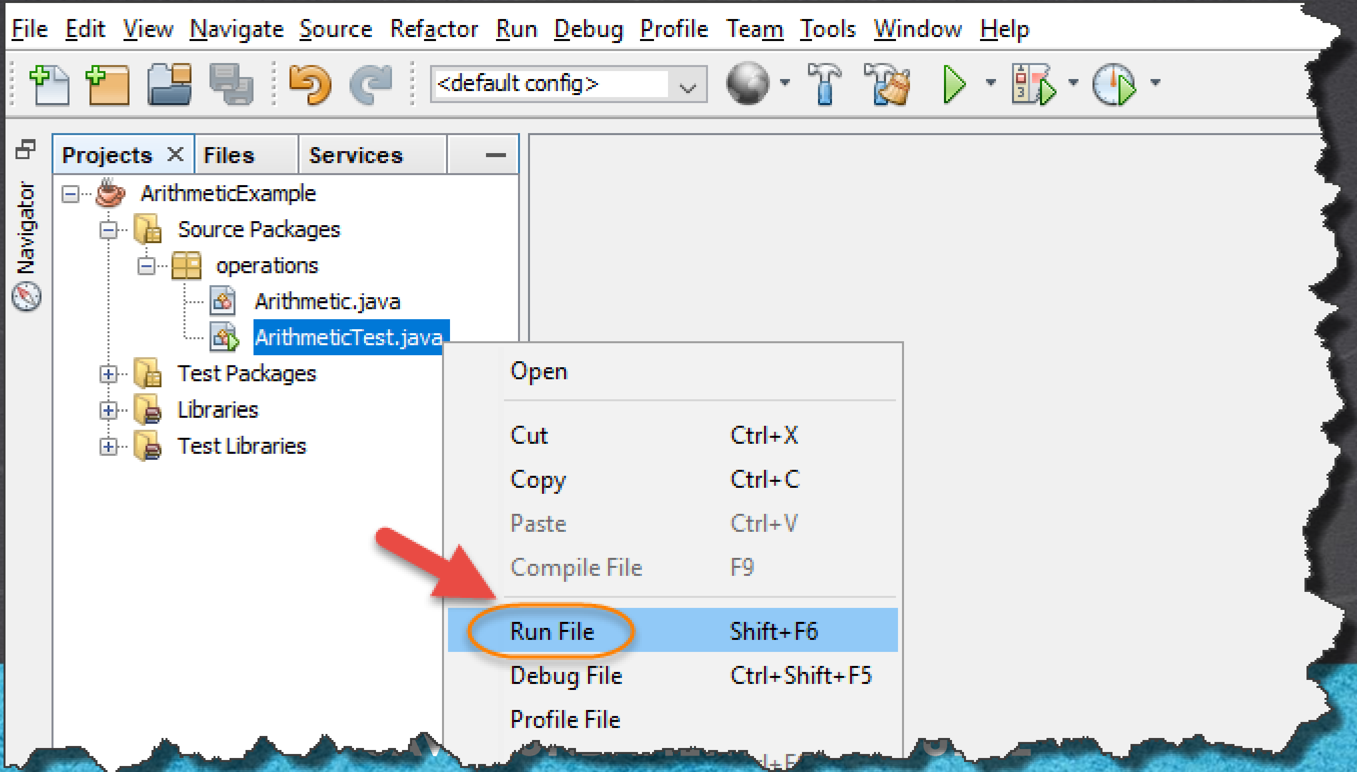
ArithmeticTest.java:



```java
package operations;

public class ArithmeticTest {

    public static void main(String args[]){

        //We create an object of the Arithmetic class with the empty constructor
        Arithmetic obj1 = new Arithmetic();

        //We call the add method and receive the returned value
        int resultado = obj1.add(5, 4);

        System.out.println("Direct use of adding without using attributes of the class. obj1:" + resultado);

        //If we call directly the method adding without arguments the value is zero,
        //since the attributes of the object were never initialized
        //since the constructor with arguments was not used, but the empty constructor
        System.out.println("\nResult of adding using attributes obj1:" + obj1.add() );

        //Now we create an object with the constructor with 2 arguments
        Arithmetic obj2 = new Arithmetic(2,1);

        //We print the result directly.
        //By directly calling the sum method, we return the value of the sum
        System.out.println("\nResult of adding using attributes obj2:" + obj2.add() );

    }
}
```

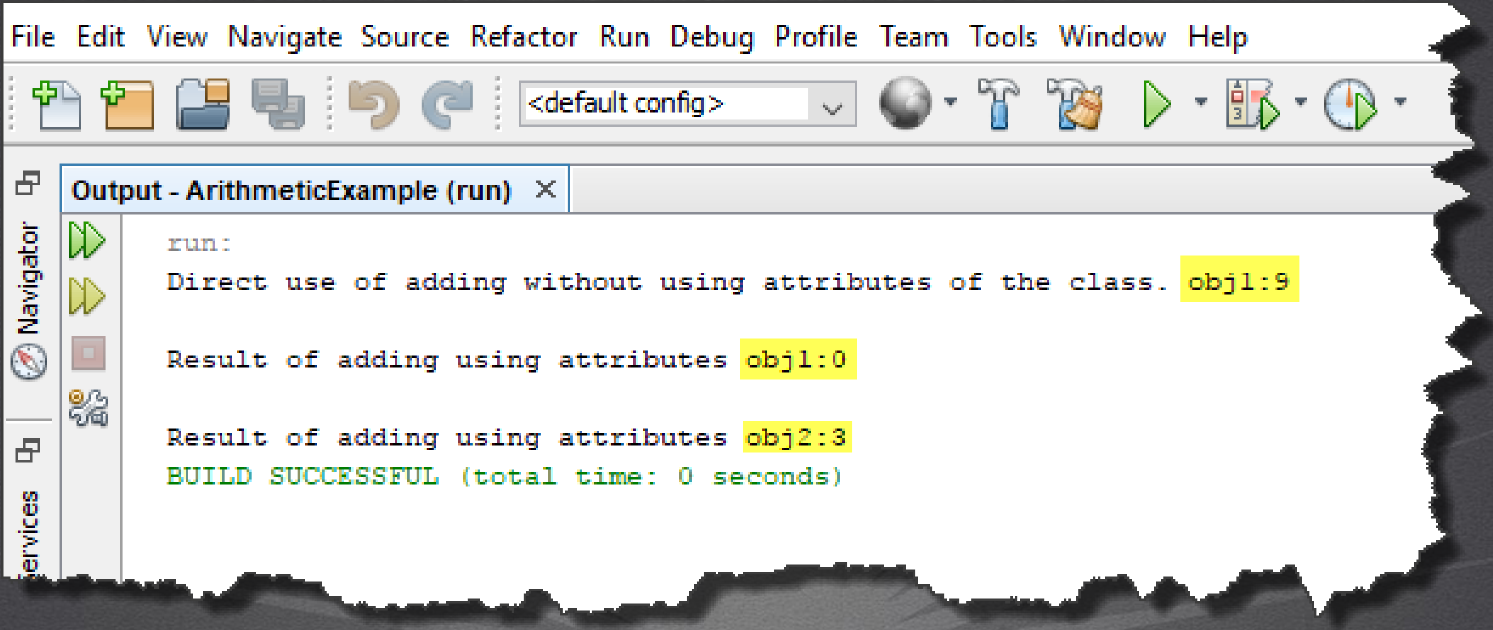
6. EXECUTE THE PROJECT

We run the project as show below:



6. EXECUTE THE PROJECT (CONT)

The result is as follows:



```
run:
Direct use of adding without using attributes of the class. obj1:9

Result of adding using attributes obj1:0

Result of adding using attributes obj2:3
BUILD SUCCESSFUL (total time: 0 seconds)
```

EXERCISE CONCLUSION

With this exercise we have implemented the use of constructors in Java.

Constructors allow us to create our objects with custom values from the moment of the creation of the object.

A Constructor is a type of method that can only be used when creating an object and must be called the same as the class.

ONLINE COURSE

JAVA FUNDAMENTALS

By: Eng. Ubaldo Acosta

Global Mentoring

JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx