

# JAVA FUNDAMENTALS COURSE

## EXERCISE

## FINAL KEYWORD



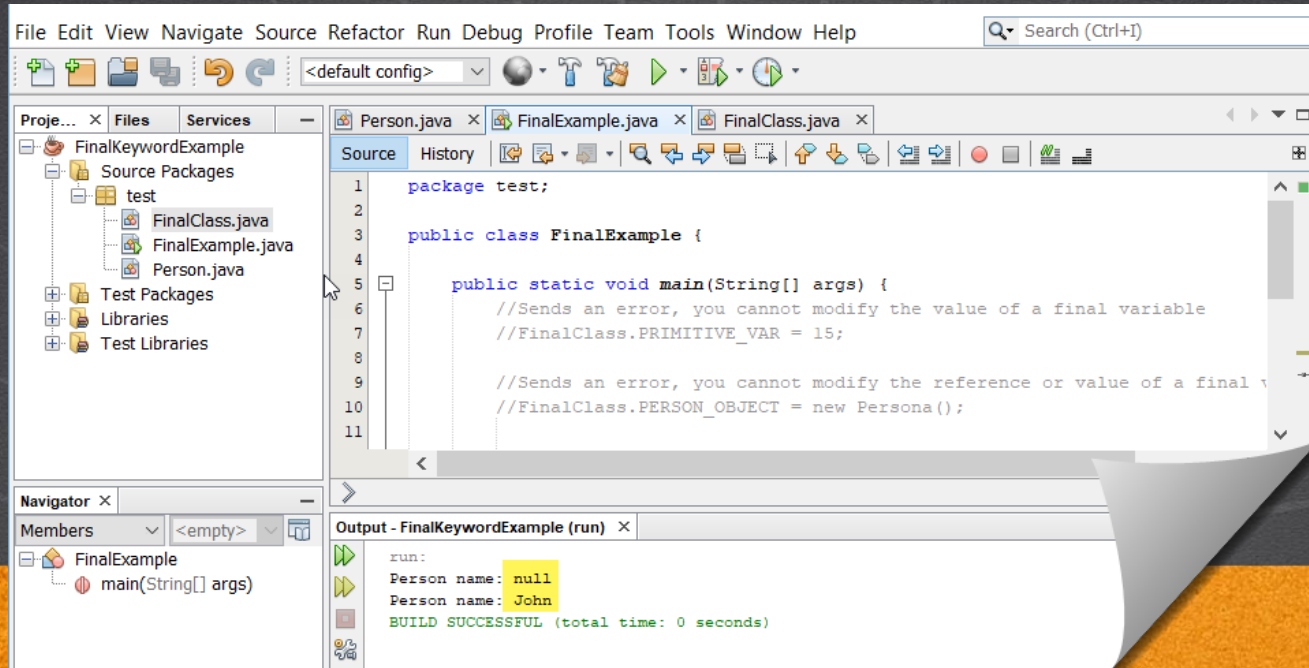
Experiencia y Conocimiento para tu vida

**JAVA FUNDAMENTALS COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# EXERCISE OBJECTIVE

Create an exercise to put into practice the use of the final word. At the end we should observe the following :





# CONVENTION

In Java if a variable is declared as **final** and static, it is considered a constant, since its value can no longer change, therefore follows the following nomenclature. The name of the variable must be written in uppercase and if the variable name contains more than one word, then each word is separated by a hyphen under. And it's recommended to declare and initialize the variable at the same time.

For example:

```
public static final int MAX_ELEMENTS = 10;
```

# 1. CREATE A NEW PROJECT

Create a new project:

**New Java Application**

**Steps**

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

**Name and Location**

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

☐ Create Main Class

**JAVA FUNDAMENTALS COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 2. CREATE A CLASS

Create a new class:

**New Java Class**

**Steps**

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

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

< Back   Next >   **Finish**   Cancel   Help

**JAVA FUNDAMENTALS COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



### 3. MODIFY THE CODE

Person.java:

```
package test;

public class Person {

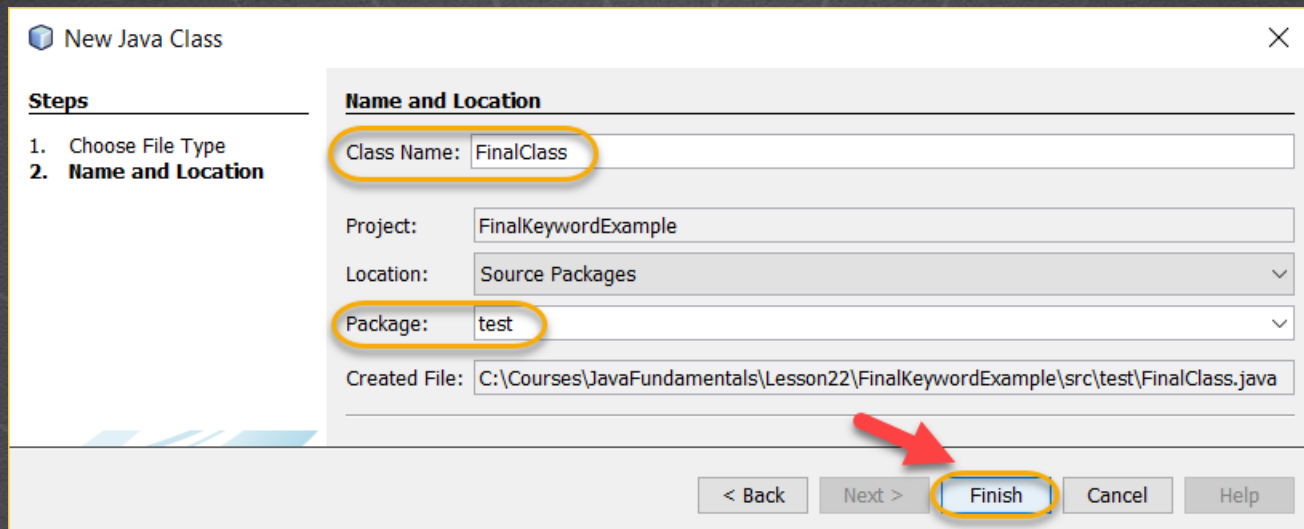
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

## 4. CREATE A CLASS

Create a class:



**New Java Class**

**Steps**

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

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

< Back   Next >   **Finish**   Cancel   Help

**JAVA FUNDAMENTALS COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# PASO 5. MODIFICAMOS EL CÓDIGO

## FinalClass.java:

```
package test;

public final class FinalClass {

    //Variables marked as final
    public static final int PRIMITIVE_VAR = 10;

    public static final Person PERSON_OBJECT = new Person();

    //final method, we will study the overwriting topic in another lesson
    public final void metodoFinal(){
    }
}

//Sends an error: Cannot inherit from a final class
//class ChildClass extends FinalClass{ }
```



## 4. CREATE A CLASS

Create a class:

**New Java Class**

**Steps**

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

**Name and Location**

Class Name: FinalExample

Project: FinalKeywordExample

Location: Source Packages

Package: test

Created File: C:\Courses\JavaFundamentals\Lesson22\FinalKeywordExample\src\test\FinalExample.java

< Back Next > **Finish** Cancel Help

**JAVA FUNDAMENTALS COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# PASO 6. MODIFICAMOS EL CÓDIGO

## FinalExample.java

```
package test;

public class FinalExample {

    public static void main(String[] args) {
        //Sends an error, you cannot modify the value of a final variable
        //FinalClass.PRIMITIVE_VAR = 15;

        //Sends an error, you cannot modify the reference or value of a final variable
        //FinalClass.PERSON_OBJECT = new Persona();

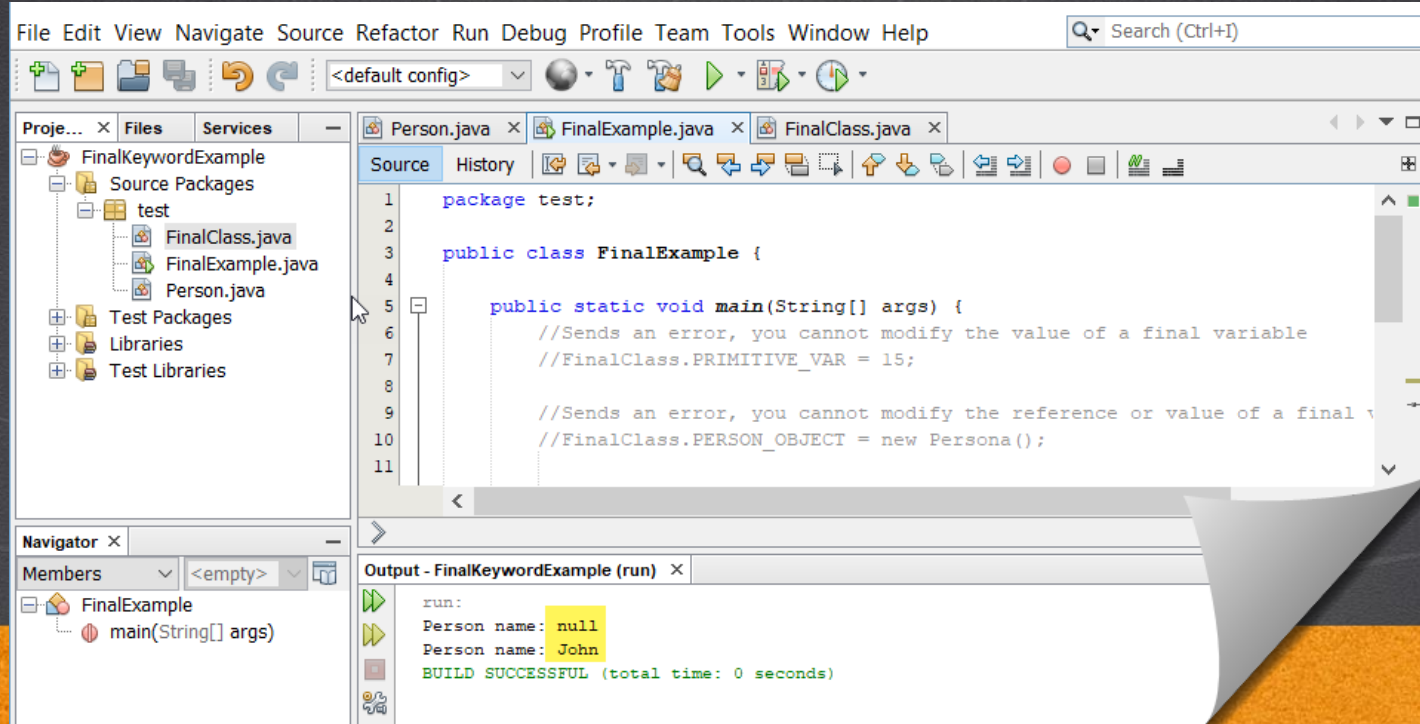
        System.out.println("Person name: " + FinalClass.PERSON_OBJECT.getName());

        //But it is possible to modify the state of the object
        //referenced by the final variable
        FinalClass.PERSON_OBJECT.setName("John");

        System.out.println("Person name: " + FinalClass.PERSON_OBJECT.getName());
    }
}
```

# 7. EXECUTE THE PROJECT

The result is as follows:





# EXERCISE CONCLUSION

- With this exercise we have put into practice the use of the final keyword.
- We have seen that it is possible to use the final keyword in 3 areas: when declaring a variable, in a method and in a class, and each one has its details. Later we will see in depth the topic of overwriting, and there we will return to the use of the final word to avoid overwriting, a concept that we will review in the next level.

**ONLINE COURSE**

# **JAVA FUNDAMENTALS**

By: Eng. Ubaldo Acosta



**JAVA FUNDAMENTALS COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)