JAVA PROGRAMMING COURSE

# EXERCISE

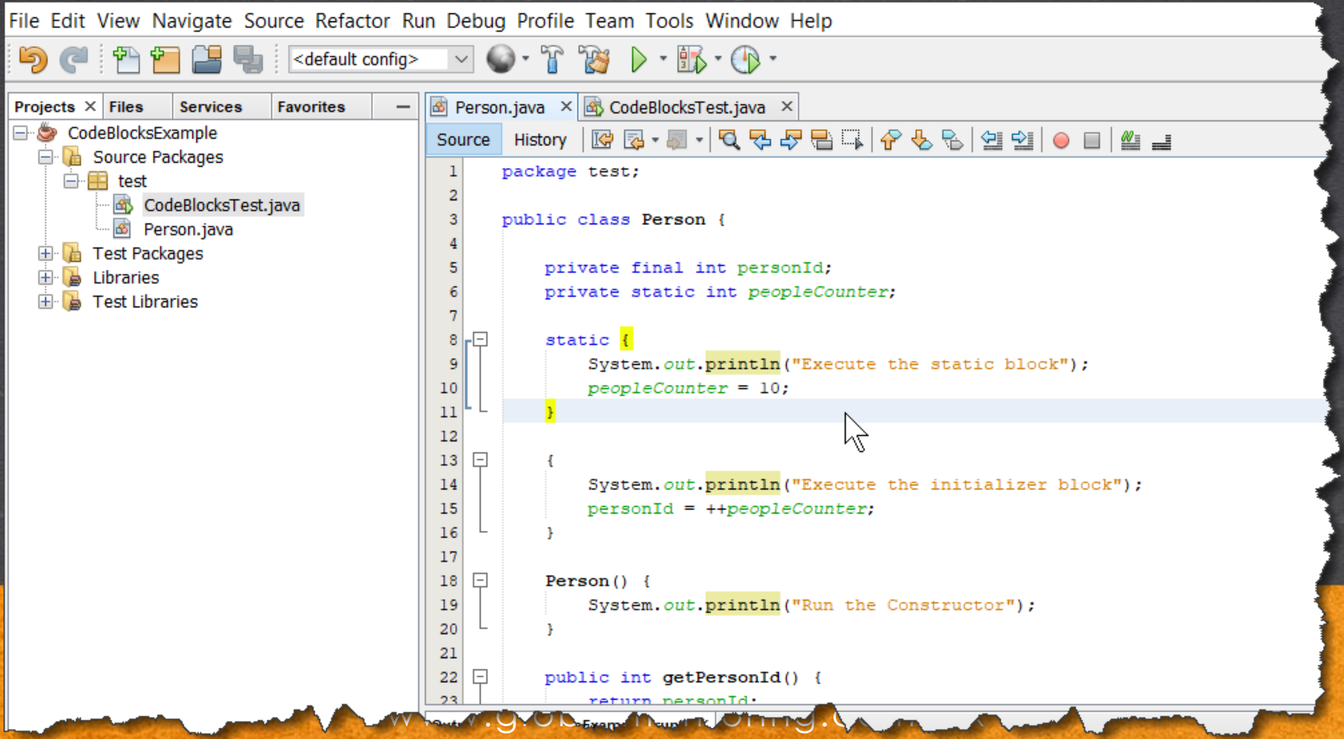
# CODE BLOCKS IN JAVA

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

# EXERCISE OBJECTIVE

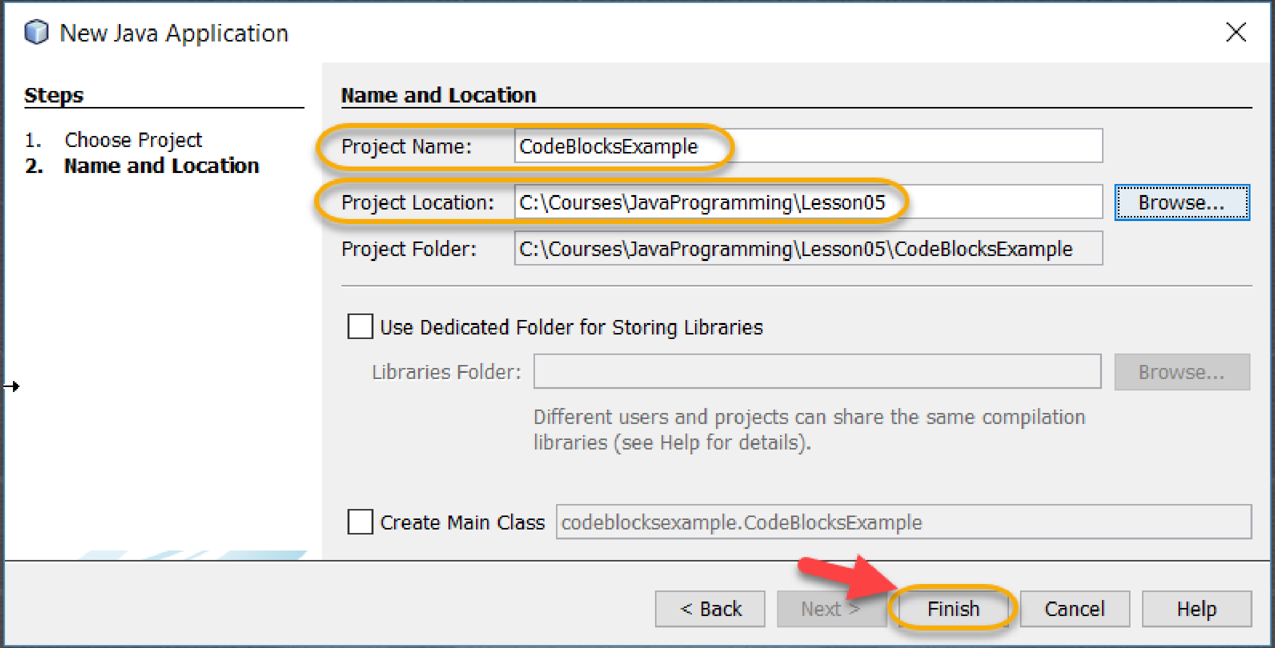
Create an exercise in the use of code blocks. At the end we should observe the following:



```java
package test;

public class Person {

    private final int personId;
    private static int peopleCounter;

    static {
        System.out.println("Execute the static block");
        peopleCounter = 10;
    }

    {
        System.out.println("Execute the initializer block");
        personId = ++peopleCounter;
    }

    Person() {
        System.out.println("Run the Constructor");
    }

    public int getPersonId() {
        return personId;
```

# 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: CodeBlocksExample

Project Location: C:\Courses\JavaProgramming\Lesson05

Project Folder: C:\Courses\JavaProgramming\Lesson05\CodeBlocksExample

☐ Use Dedicated Folder for Storing Libraries

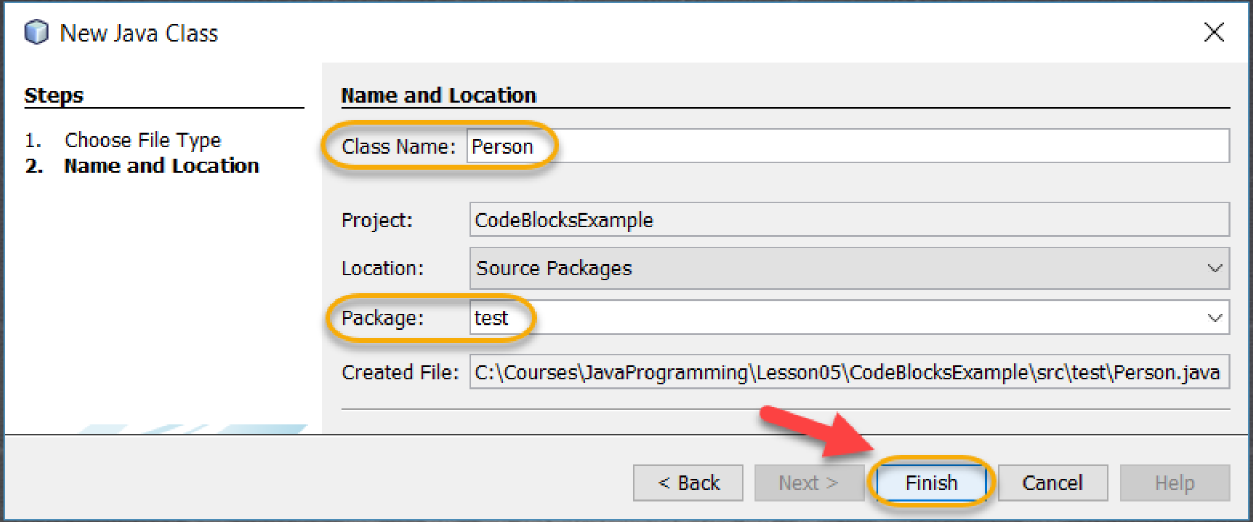
Libraries Folder:

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

☐ Create Main Class codeblocksexample.CodeBlocksExample

# 2. CREATE A NEW CLASS

Create a new class:



New Java Class

**Steps**

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

**Name and Location**

Class Name: Person

Project: CodeBlocksExample

Location: Source Packages

Package: test

Created File: C:\Courses\JavaProgramming\Lesson05\CodeBlocksExample\src\test\Person.java

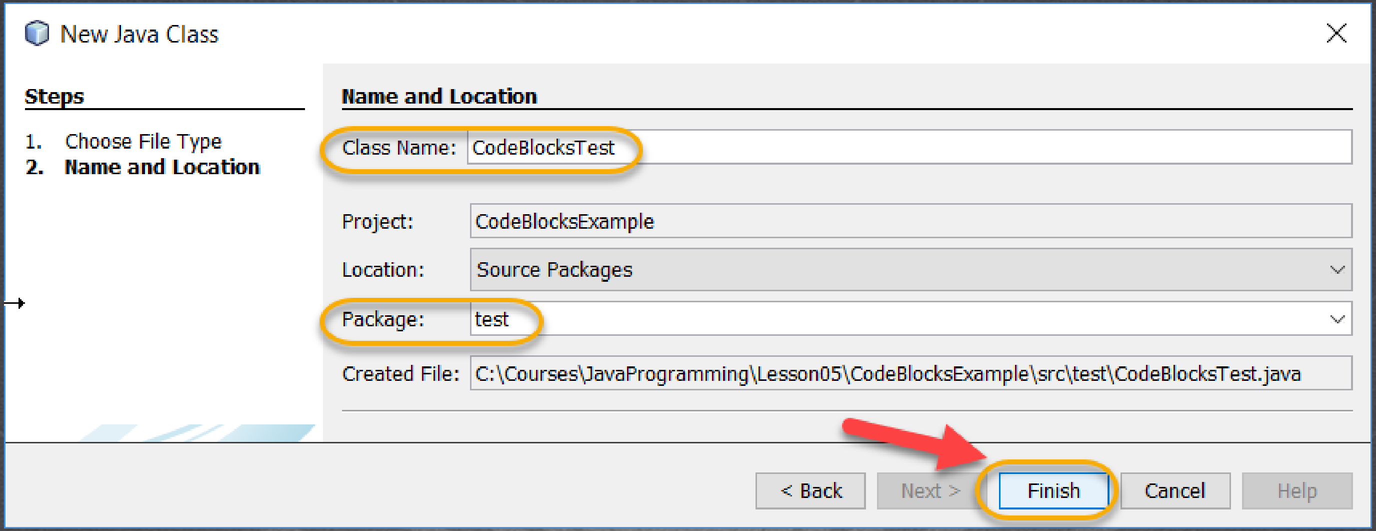
# 3. MODIFY THE CODE

## Person.java:

```java
package test;

public class Person {

    private final int personId;
    private static int peopleCounter;

    static {
        System.out.println("Execute the static block");
        peopleCounter = 10;
    }

    {
        System.out.println("Execute the initializer block");
        personId = ++peopleCounter;
    }

    Person() {
        System.out.println("Run the Constructor");
    }

    public int getPersonId() {
        return personId;
    }

}
```

# 4. CREATE A NEW CLASS

Create a new class:



New Java Class

**Steps**

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

**Name and Location**

Class Name: CodeBlocksTest

Project: CodeBlocksExample

Location: Source Packages

Package: test

Created File: C:\Courses\JavaProgramming\Lesson05\CodeBlocksExample\src\test\CodeBlocksTest.java

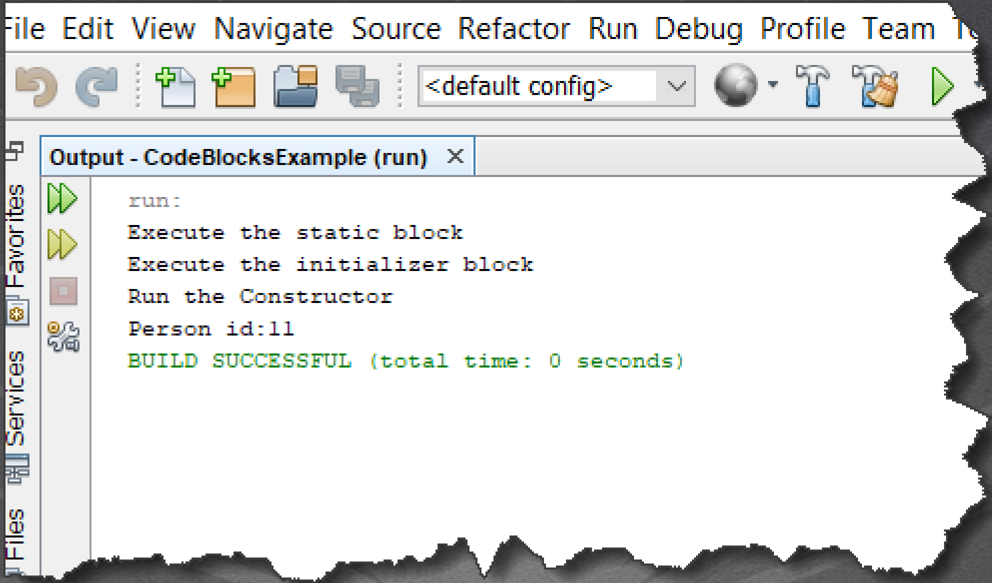
# 4. MODIFY THE CODE

## CodeBlocksTest.java:

```java
package test;

public class CodeBlocksTest {

    public static void main(String[] args) {

        Person p1 = new Person();
        int id = p1.getPersonId();
        System.out.println("Person id:" + id);
    }

}
```

## 5. EXECUTE THE PROJECT

The result is as follows:


File Edit View Navigate Source Refactor Run Debug Profile Team
<default config>
Output - CodeBlocksExample (run)
Favorites
Services
Files

```
run:
Execute the static block
Execute the initializer block
Run the Constructor
Person id:11
BUILD SUCCESSFUL (total time: 0 seconds)
```

## EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of code blocks, we have seen how to use them to initialize variables, both static and non-static.

- These blocks will be used to initialize variables that may need more complex processes to assign the value, or we simply do not have yet the value to assign to the variable. For more information you can see:
- https://docs.oracle.com/javase/tutorial/java/javaOO/initial.html

ONLINE COURSE

# JAVA PROGRAMMING

By: Eng. Ubaldo Acosta

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx