JAVA FUNDAMENTALS COURSE

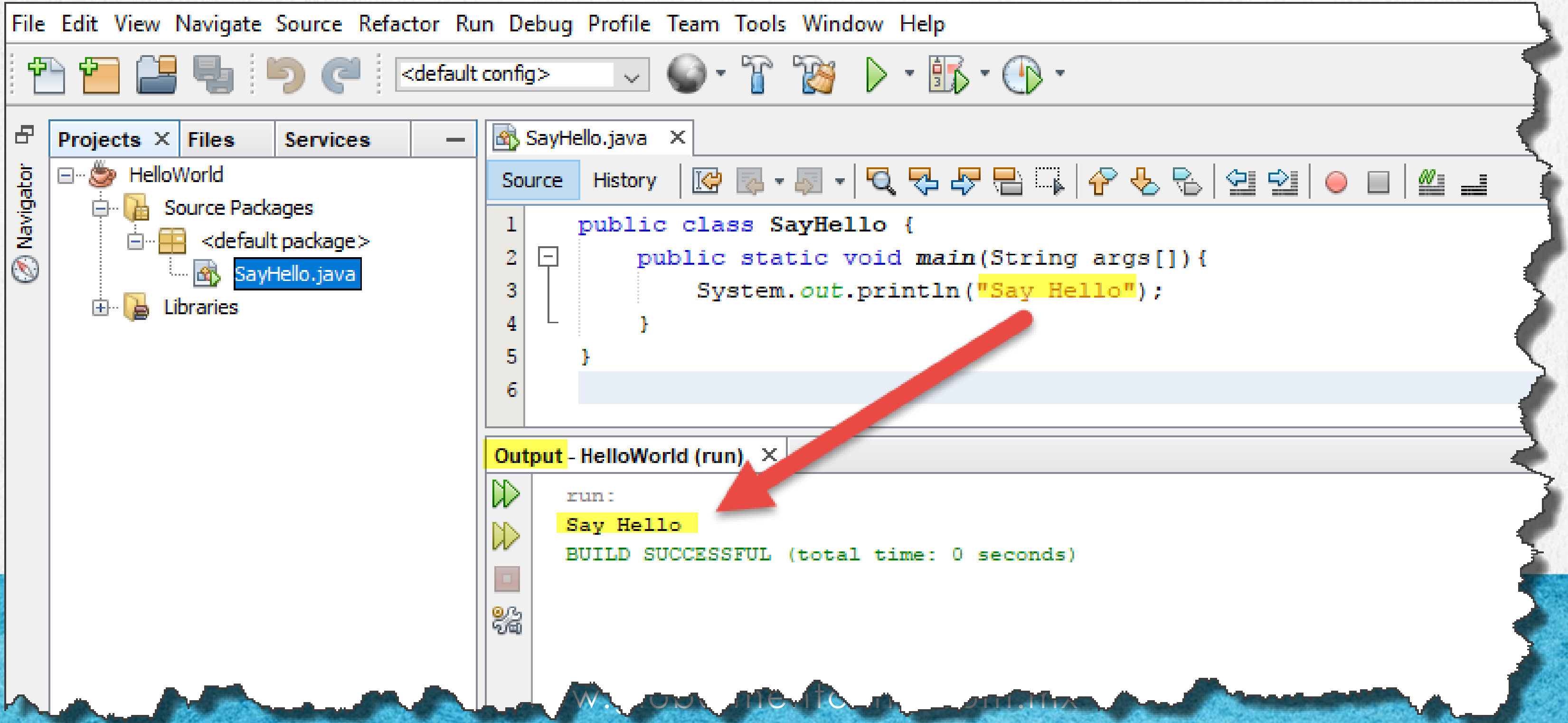
EXERCISE

HELLOWORLD WITH JAVA AND NETBEANS

Global Mentoring

EXERCISE OBJECTIVE

In this first exercise, we are going to create our first Java program using Netbeans. At the end we should see the next result:



```java
public class SayHello {
    public static void main(String args[]){
        System.out.println("Say Hello");
    }
}
```

```
run:
Say Hello
BUILD SUCCESSFUL (total time: 0 seconds)
```

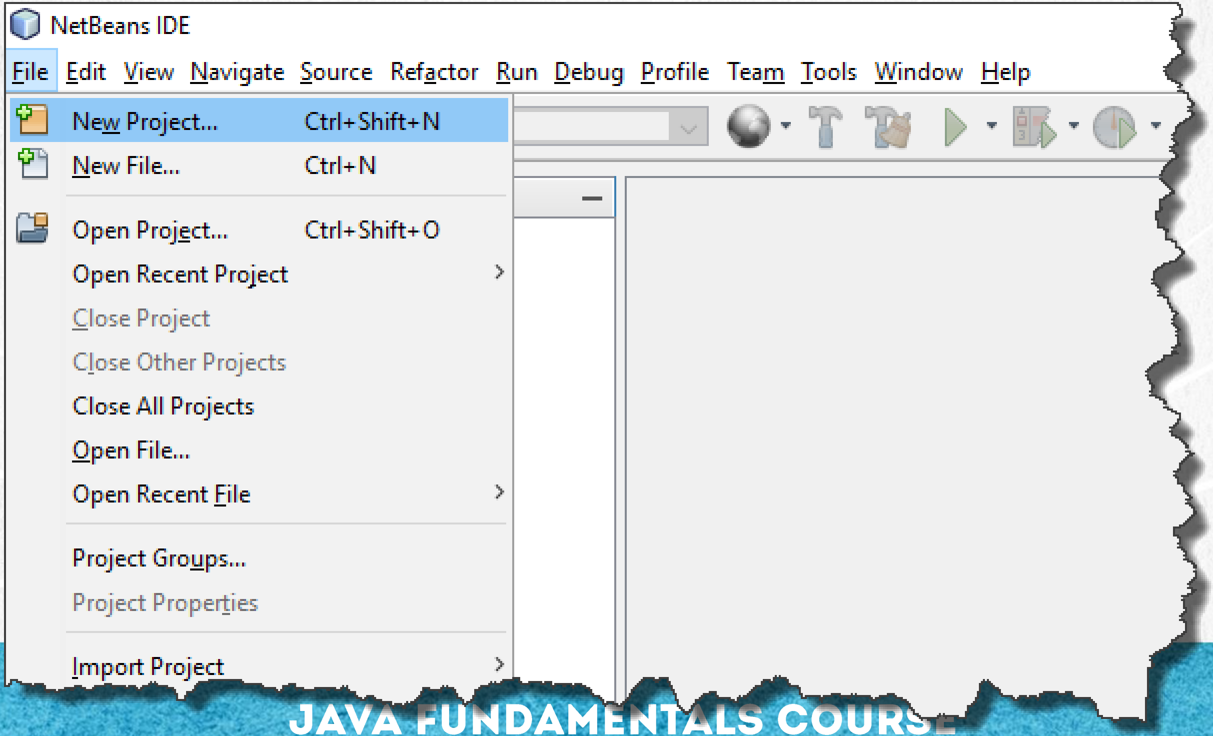
METHODOLOGY OF THE EXERCISES

Every exercise has been made step by step. So the only thing you have to do is to follow every step, and as you continue with the course you will be more familiar with our new methodology called Speed Learning, so you can study from the video, or right in the PDF or eBook file, because you have all the instructions step by step in both: the videos or the pdf files (eBooks).

We hope you enjoy our methodology. We are sure you will save tons of hours of study once you get familiarize with our Speed Learning Methodology. Let’s start this exercise.

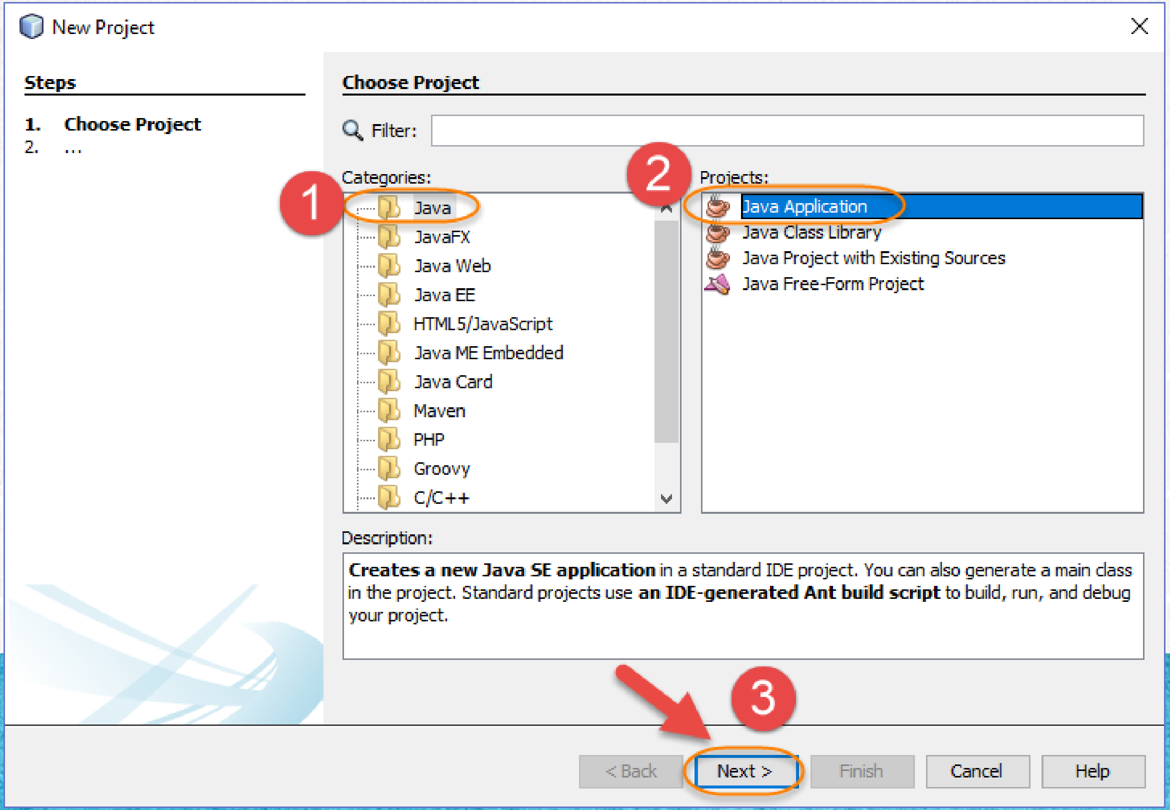
1. CREATE A NEW PROJECT

Create a new HelloWorld project:



1. CREATE A NEW PROJECT (CONT)

We select the Java category and a Java Application:



New Project

Steps

1. **Choose Project**
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- HTML5/JavaScript
- Java ME Embedded
- Java Card
- Maven
- PHP
- Groovy
- C/C++

Projects:

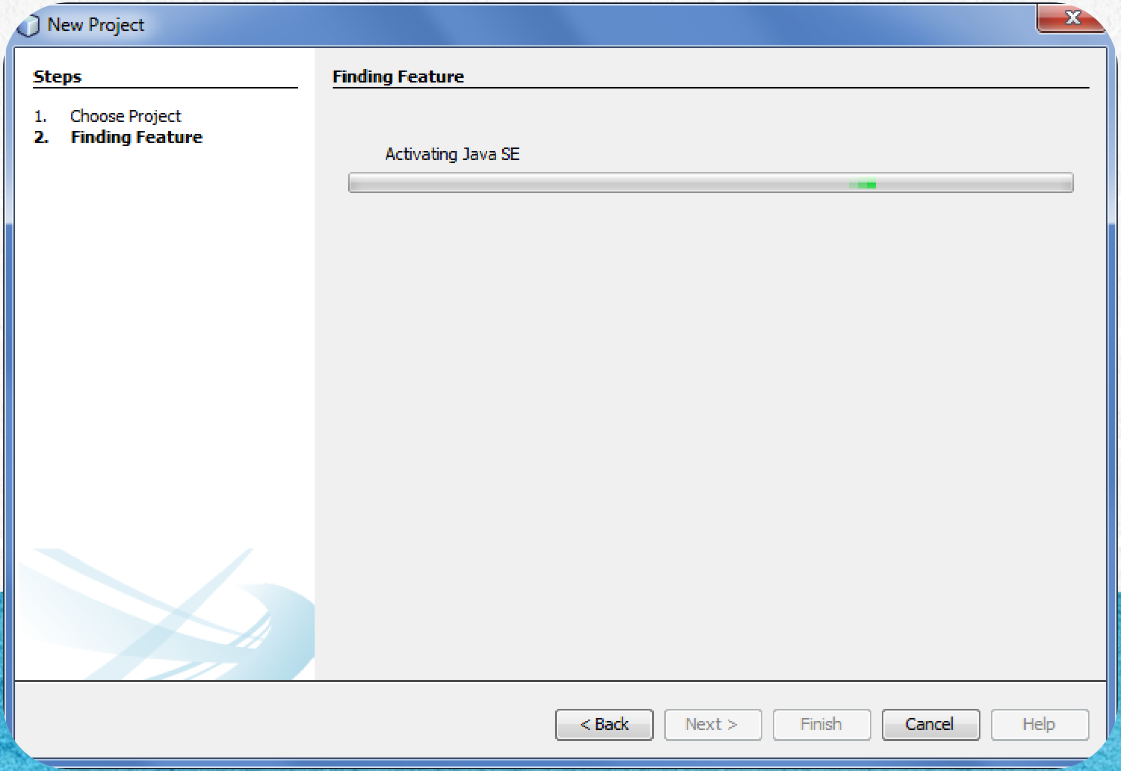
- Java Application
- Java Class Library
- Java Project with Existing Sources
- Java Free-Form Project

Description:

Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

1. CREATE A NEW PROJECT (CONT)

The first time a Java Project is created, Netbeans activates this kind of Project in Netbeans. So we wait it to finish.



New Project

Steps

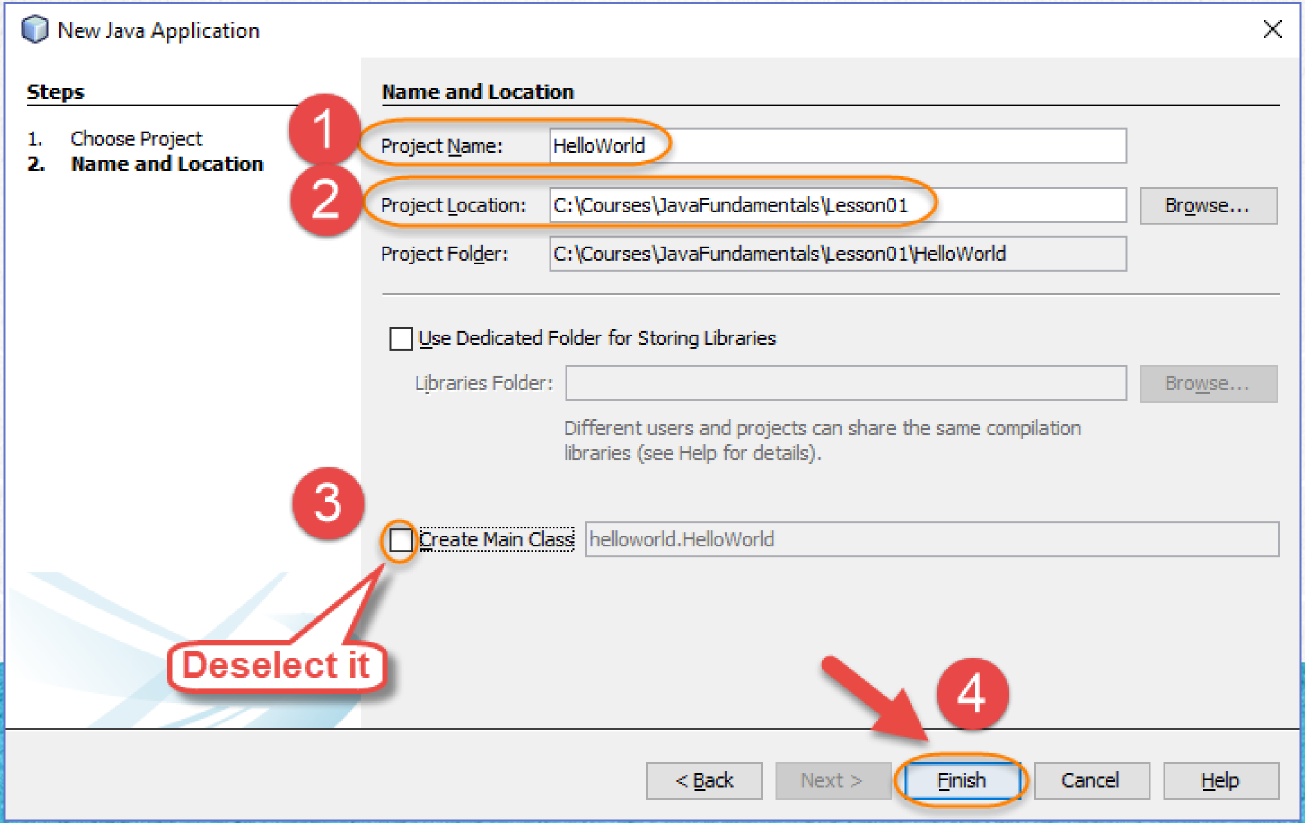
1. Choose Project
2. **Finding Feature**

Finding Feature

Activating Java SE

1. CREATE A NEW PROJECT (CONT)

We assign a name and a path for our new project as show below:



New Java Application

Steps

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

Name and Location

1 Project Name: HelloWorld

2 Project Location: C:\Courses\JavaFundamentals\Lesson01

Project Folder: C:\Courses\JavaFundamentals\Lesson01\HelloWorld

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

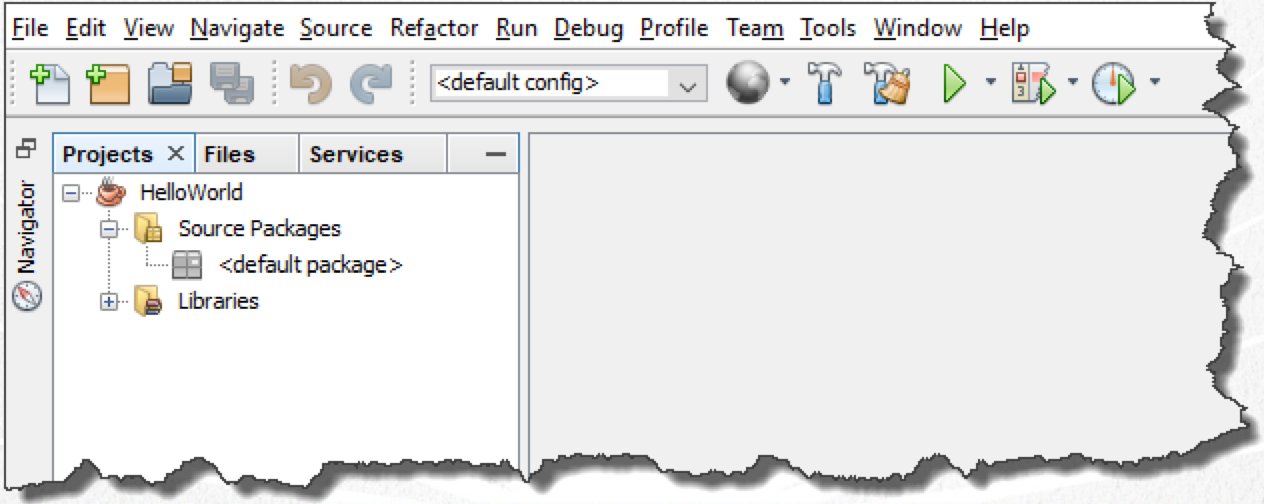
3 □ Create Main Class helloworld.HelloWorld

Deselect it

4

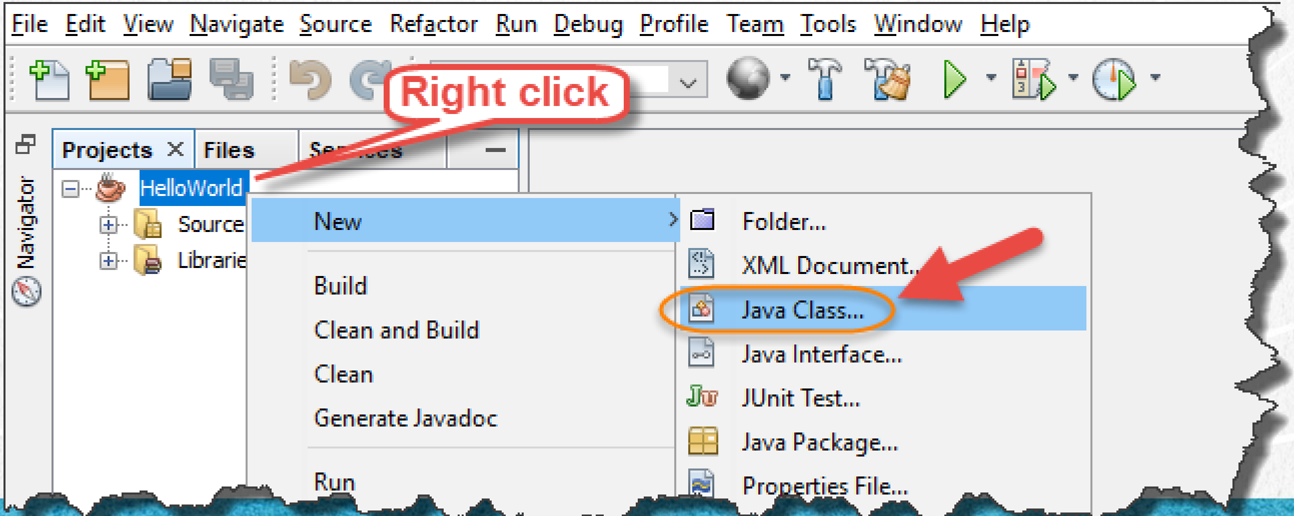
1. CREATE A NEW PROJECT (CONT)

This is the result of creating a new project. The Source Packages folder is where our Java code will live:



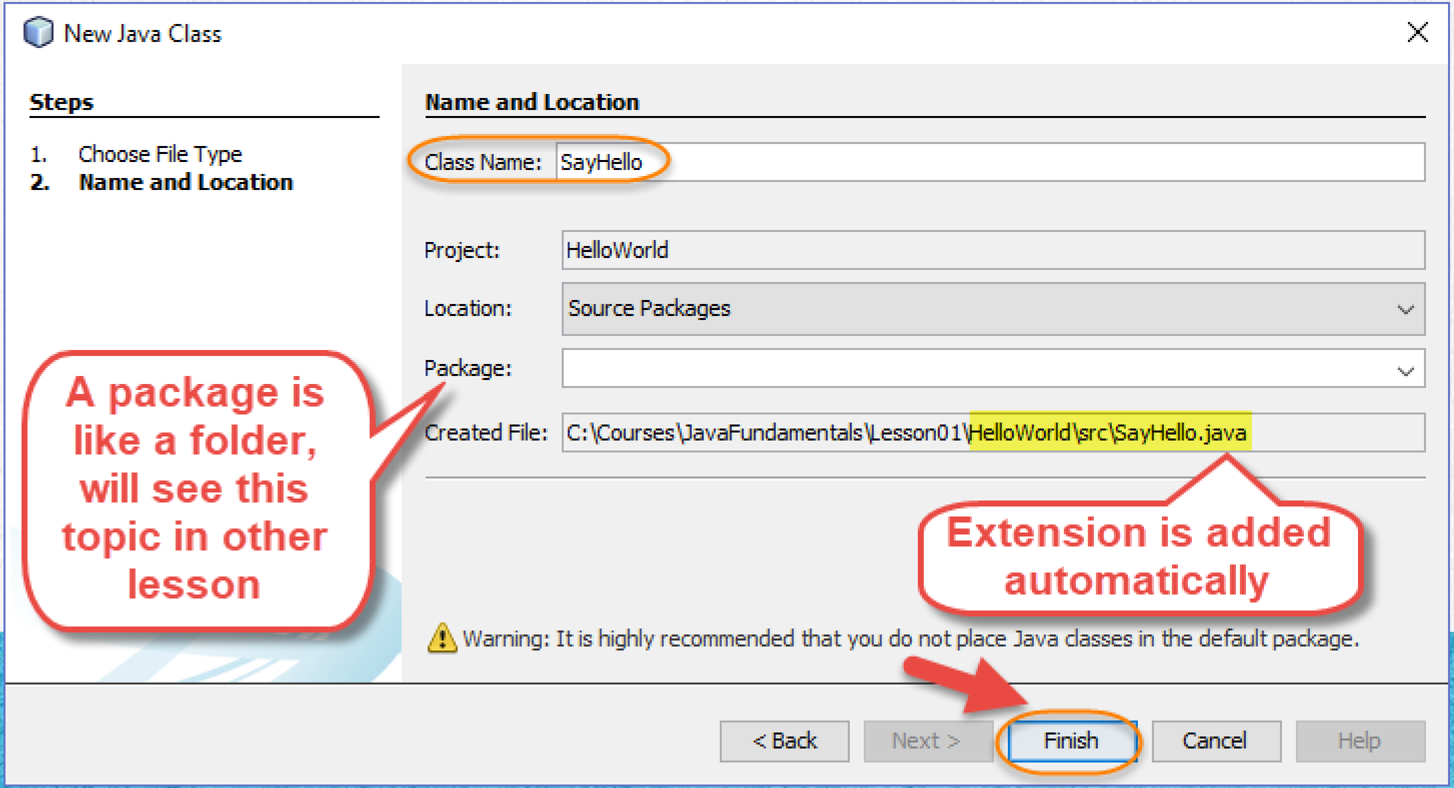
2. CREATE A JAVA CLASS

We will create a new Java class. A class is a new file with the same name in the source code and in the name of the file. The file must end with the .java extension. Let's create a new class:



2. CREATE A JAVA CLASS (CONT)

We assign the name SayHello to our new class. And click on finish:



New Java Class

Steps

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

Name and Location

Class Name: SayHello

Project: HelloWorld

Location: Source Packages

Package:

Created File: C:\Courses\JavaFundamentals\Lesson01\HelloWorld\src\SayHello.java

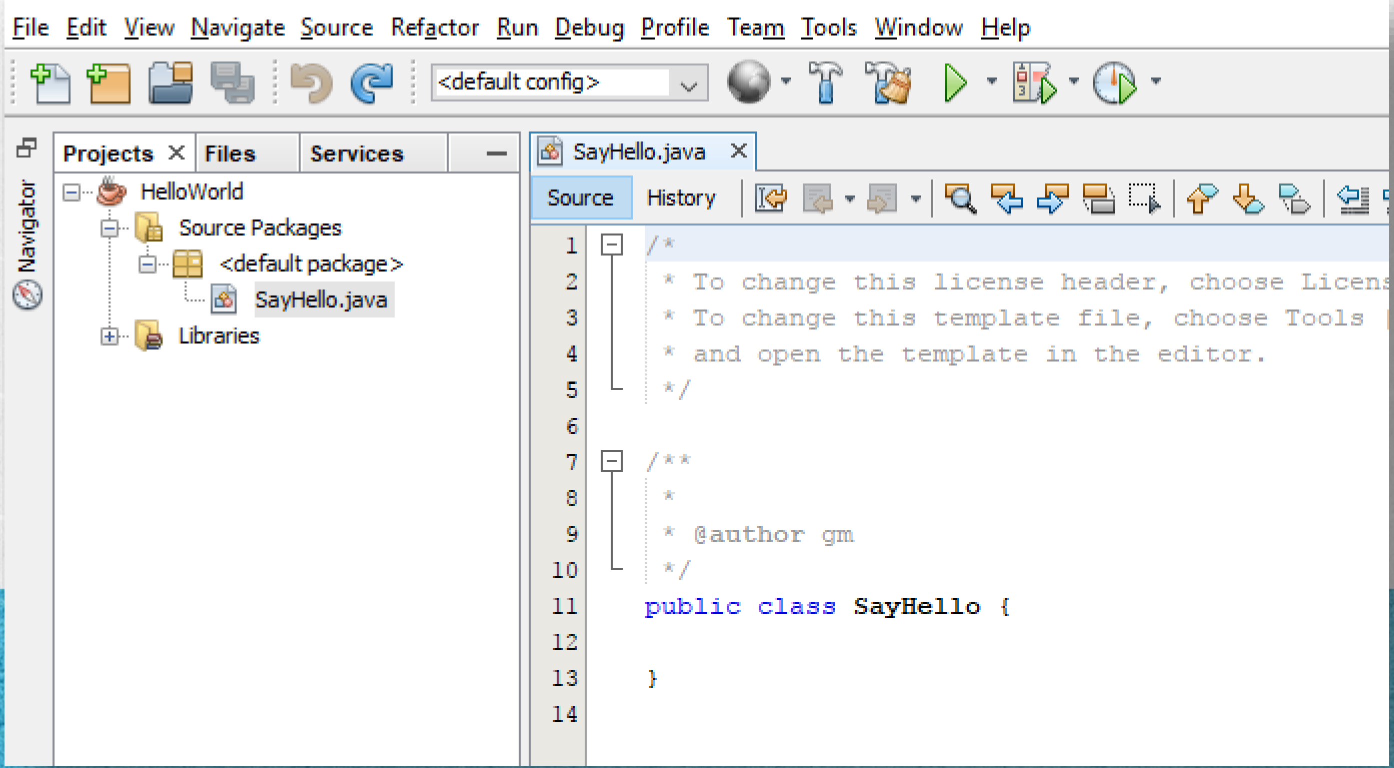
⚠ Warning: It is highly recommended that you do not place Java classes in the default package.

A package is like a folder, will see this topic in other lesson

Extension is added automatically

2. Create a Java Class (cont)

The new class must look like show below. By now, a new class must have the public modifier, the class keyword and the name of the class, in this case SayHello, and the code of the class must be inside the brackets { }. The extension .java is added automatically to the file:



```
/*
 * To change this license header, choose Licens
 * To change this template file, choose Tools 
 * and open the template in the editor.
 */

/**
 *
 * @author gm
 */
public class SayHello {

}

```

3. MODIFY THE JAVA CODE

The next step is to modify our Java code of the SayHello class. Is important to tell you that every exercise has an eBook associated, and you can download it and check the source code of every file. This is part of our Speed Learning methodology. You can always check the complete source code of every file in every exercise.

If you have problems coding the source code, you can click in the name of the file/class that is given to you, or you can download the complete Project that is 100% functional.

Let’s see how the SayHello class is implemented.

3. MODIFY THE CODE (CONT)

Archivo SayHello.java:

```java
public class SayHello {
    public static void main(String args[]){
        System.out.println("Say Hello");
    }
}
```

3. MODIFY THE CODE (CONT)

In the last code, we’ve created a class called SayHello.

And after the definition of that class, we’ve created a main method. This method is necessary to execute the class SayHello.

To define a main method that is executable, we need to write the code below:

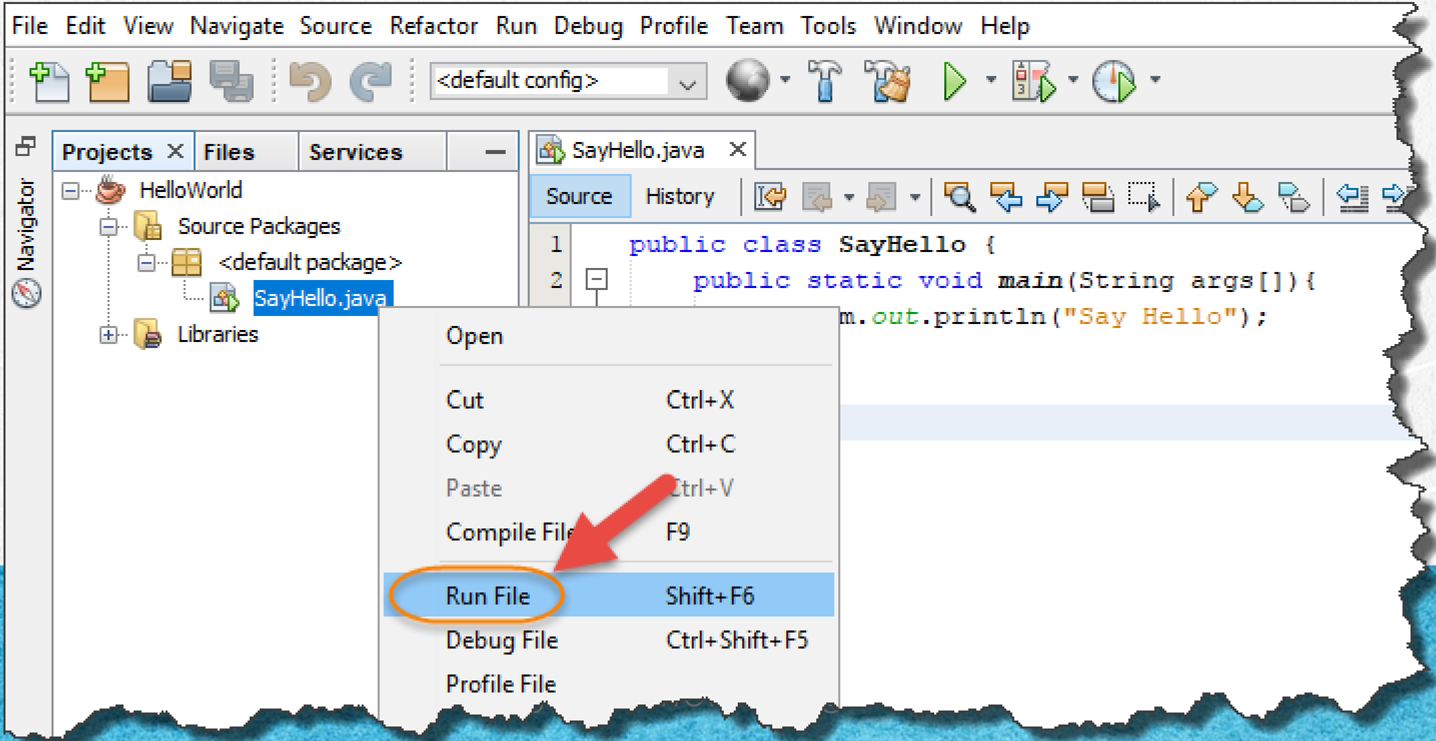
```
public static void main(String[] args){ … }
```

Let’s explain the main method:

1) public: is a Modifier, we’ll see the accessing modifiers later in the course.
2) static: is a modifier, we’ll see the static modifier later in the course.
3) void: This means the method return nothing, we’ll see this reserved word later in the course.
4) main: this is the name of the method, and must be in lower-case letter.
5) (String[] args): This is the parameter of the method, and is an array of String types. We’ll see String types and Arrays later in the course.
6) { … } This is the body of the method where the logic of the method goes.
7) Inside the body of the method there is a **System.out.println( )** method. The **println( )** method prints to the console the text or String that we pass as a parameter to this method.

4. EXECUTE THE PROJECT

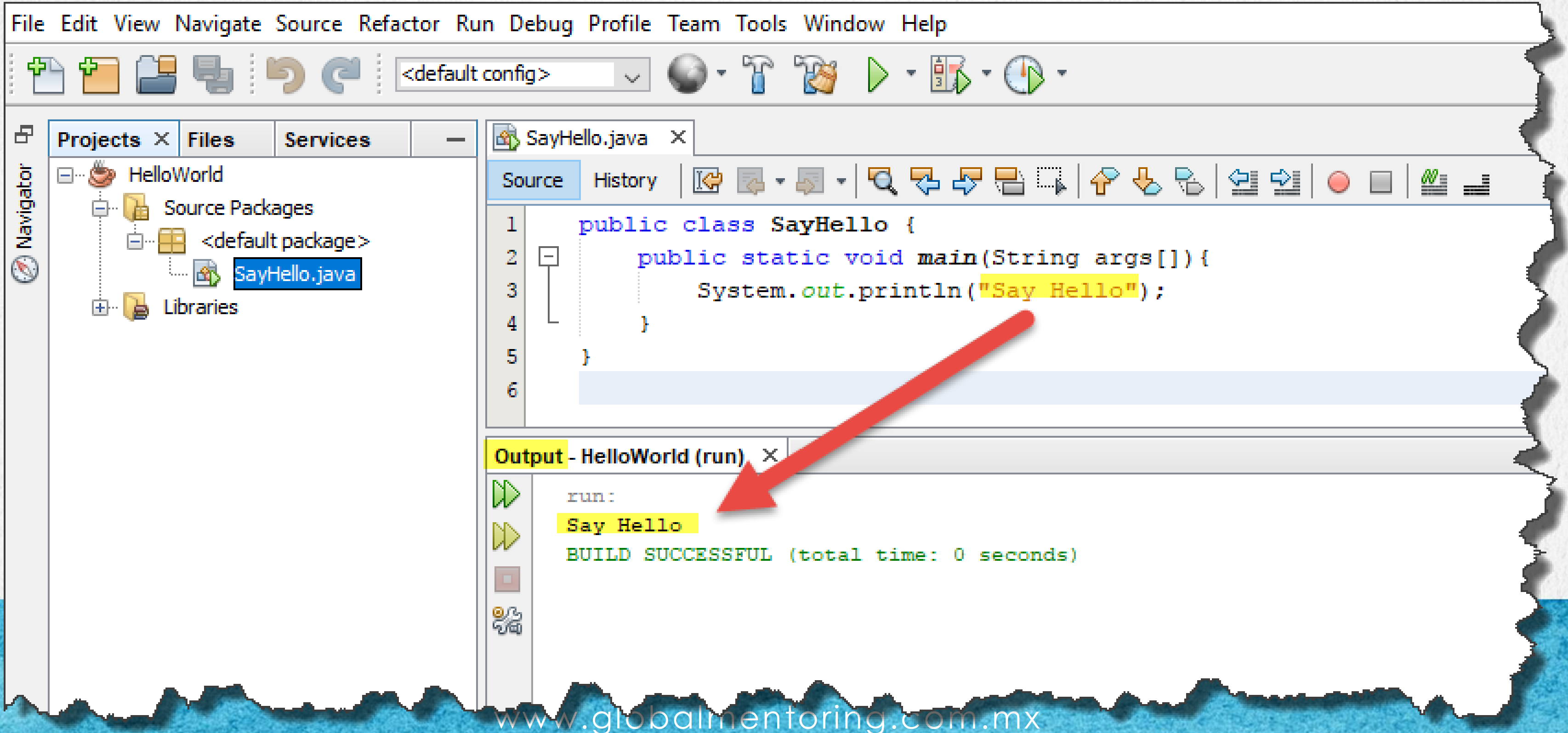
Finally, we execute our project. Right click on the SayHello class, and select "Run File":



```
public class SayHello {
    public static void main(String args[]){
            m.out.println("Say Hello");
```

4. EXECUTE THE PROJECT (CONT)

This is the result of executing our code:



```
public class SayHello {
    public static void main(String args[]){
        System.out.println("Say Hello");
    }
}
```

```
run:
Say Hello
BUILD SUCCESSFUL (total time: 0 seconds)
```

EXERCISE CONCLUSION

- With this exercise we have created our first Java program with the help of the Netbeans IDE.

- An IDE (Integrated Development Environment) is a tool that will help us to speed up our creation of software with Java, however we should not depend 100% on this type of tools, since in a job many times we will not be able to select the IDE with which we will develop the Java code, and therefore we can stay without any knowledge when changing the IDE.

- For this reason we will learn the concepts, which can be applied in any IDE regardless of which one we use or that of our preference. For this, we will provide all the fully functional code in any IDE, and so they can use the IDE of their choice.

- Finally, we know many of the concepts are new to many programmers, but believe us, we've structured the course in a way that every lesson has an specific topic and nothing more tan that, in order to retain the concept, or be able to review an concept during the course. So, please be patient when studying this course, and if you plan to be a good programmer, this is a necessary skill, so again: please be patient, you'll be an excellent Java programmer at the end of this course ☺

ONLINE COURSE

JAVA FUNDAMENTALS

Author: Ubaldo Acosta

Global Mentoring