

Week 01 5COSC019W – Object Oriented Programming – Java

Refresh some Java concepts and the use of NetBeans

Week 01: 23/09/2024 – 27/09/2024

There are no tutorials on Week 1, but if you would like to revise some Java concepts, you can attempt the exercises below.

The solution code is provided on BB.

In this exercise, we will refresh some essential Java programming concepts that you know!

Be sure you are able to implement in Java the following:

- If statements
- For/while loops
- Methods implementation
 - o Parameter passing
 - o Returned value
- Arrays

We will refresh these concepts in this first tutorial.

If you don't feel confident with basic Java, you are strongly recommended to revisit the module "Software Development II".

Setting Up the Project – Java Application

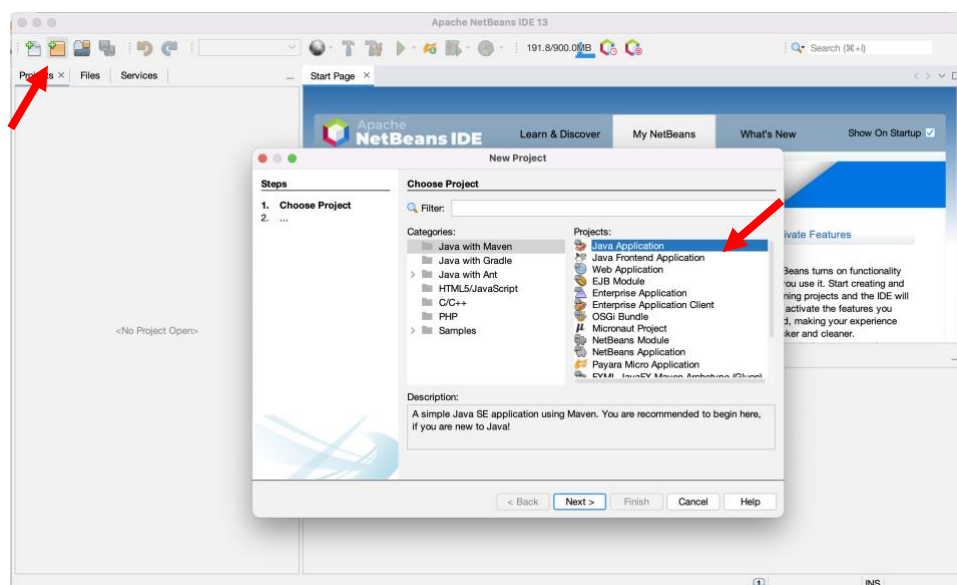
During the lectures and tutorials, we will be using **NetBeans 13**. This is strongly recommended, as your Lab-based practical test will be in NetBeans 13.

This version is installed on all the university computers, and you can access them remotely from home using Splashtop. On the module Blackboard site, you will find guidance on how to install Splashtop and access the computers remotely.

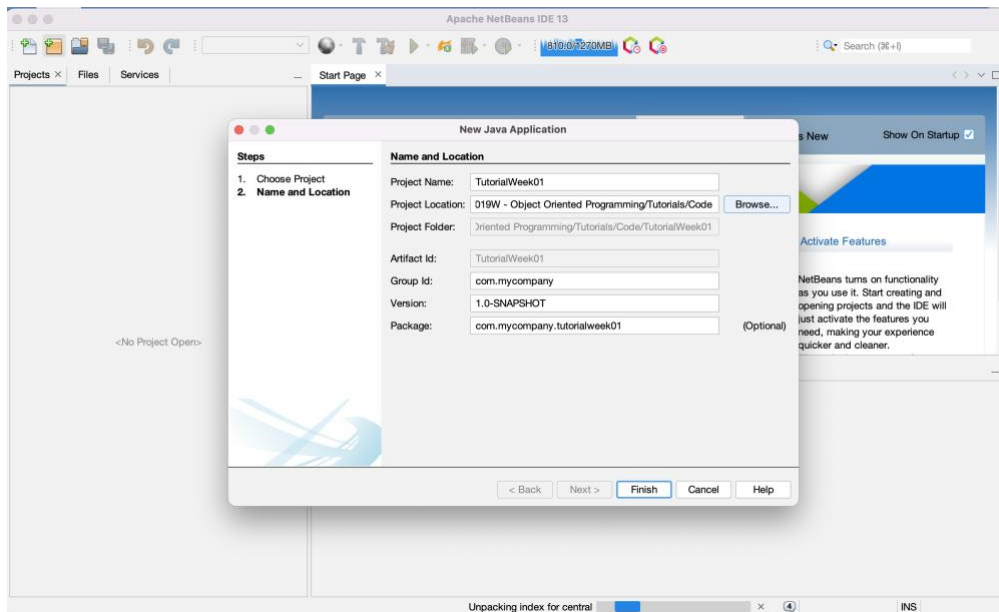
1) Start NetBeans 13 IDE.

Select New Project.

In the New Project wizard, expand the category "Java with Maven" and select "Java Application" as shown in the figure below. Then click Next

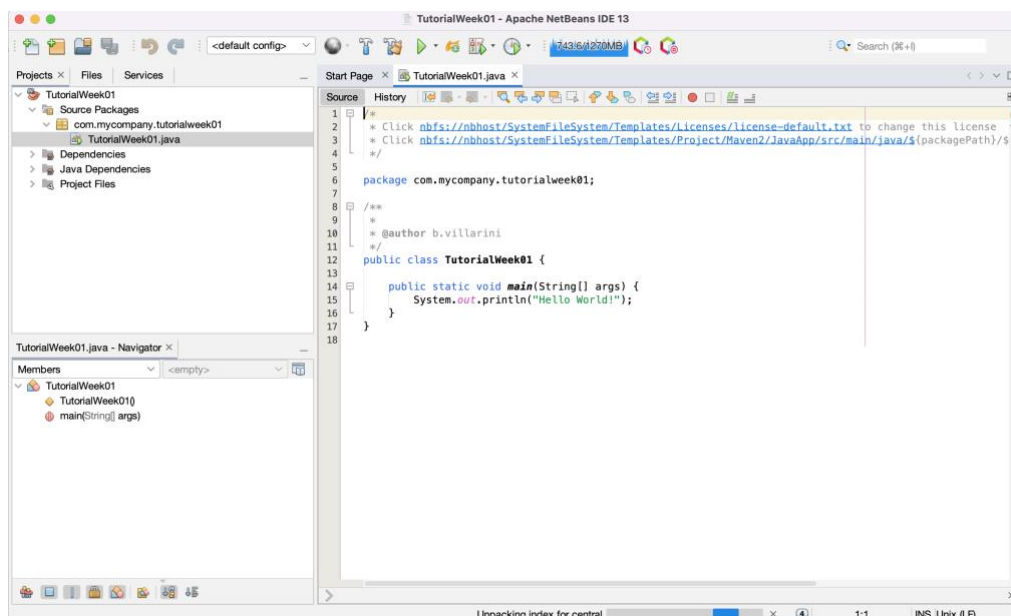


- 2) In the Name and Location page of the wizard, do the following (as shown in the figure below):
- In the Project Name field, type *TutorialWeek01*.
 - In the Project Location field, choose the folder where to save your project
 - You can leave the other fields as they are
 - Click finish



The project is created and opened in the IDE. You should see the following components:

- The **Projects window** contains a tree view of the components of the project, including source files, libraries that your code depends on, and so on.
- The **Source Editor** window with a file called *TutorialWeek01.java*.
- The **Navigator window** is used to quickly navigate between elements within the selected class.

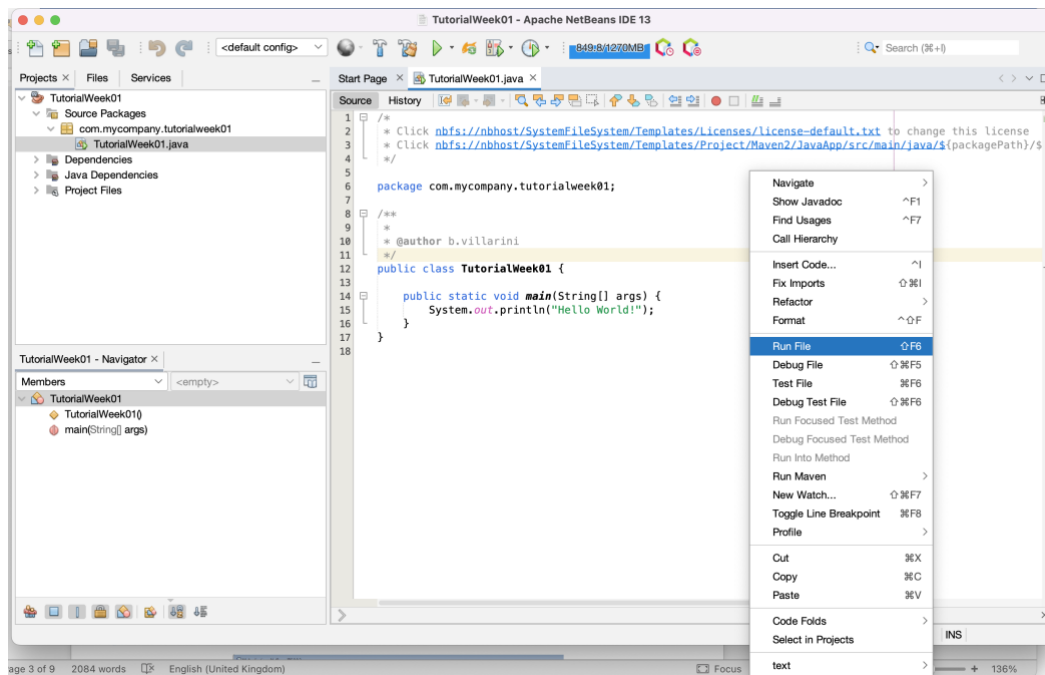


A skeleton code has been automatically generated:

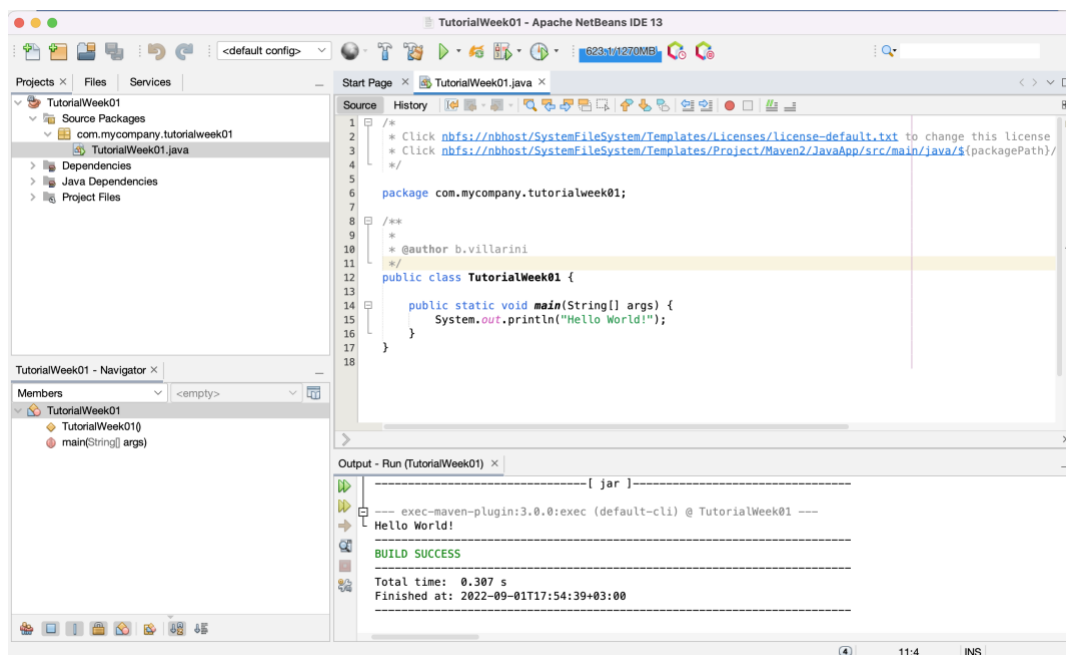
- A class called *TutorialWeek01* has been created
- A main method has been created
- Within the body of the main method, there is an instruction to print "Hello World!" on the screen.

3) Run the code:

- right click in the file and then select “Run” (or you can press the green triangle on the toolbar)



- you will see that the code compiled successfully and in the output window you will find the text “Hello world!”



Implement a program to compute the average of a set of marks

Your program should be able to do the following

- 1) The user will insert six marks from the keyboard. Save these marks in an array of integers and print them back on the screen.

Some hints... The instructions you should implement are:

- Declare a Scanner object to read an integer from the keyboard
- Declare an array of integers to save the six marks (length of the array = 6)
- Implement a loop to iterate the action to enter the mark and save it into the array
- Print all the marks in the array

- 2) Print how many marks are below 40 (Fail)
Print how many marks are from 40 to 49 (Pass)
Print how many marks are from 50 to 59 (2:2)
Print how many marks are from 60 to 69 (2:1)
Print how many marks are equal or above 70 (First)

Some hints... You can use a set of if statements (or a switch case) to check which condition is true and then print the output on the screen.

- 3) Write a method that computes the average among the six marks.
The method should take as a parameter an array of integers and return the average as a double.
From the main method, you call this method, passing the array with your six marks and printing the output on the screen.
- 4) Write a method that computes the maximum mark among those saved in the array and prints it on the screen. The method should take as a parameter an array of integers and return an int.
- 5) Write a method that computes the minimum mark among those saved in the array and prints it on the screen. The method should take as a parameter an array of integers and return an int.
- 6) Write a method that sorts the marks in the array in ascending order (from the smallest to the largest).
The method should take as a parameter an array of integers and return an array of integers.