

Lab Exercise 1 (**Bonus**): Revisiting Java Fundamentals – Part 1

SE2205: Data Structures and Algorithms using Java – Fall 2021

Open Day: September 12, 2021; **Cut off time:** Friday September 17, 2021 @11pm

Adapted by Dr. Quazi Rahman (qrahman3@uwo.ca).

A. Rationale and Background

In this lab exercise we will review basic programming concepts in the context of Java, using two problems.

B. Evaluation and Submission Instructions

You will get **BONUS** credit for this lab exercise when you submit the working code. No part-mark will be awarded if the code does not run. Submit your lab work online by carrying out the following Instructions:

1. Create a Project with the following name: *username_Lab1*
2. For each question create a package (For Q1: Q1L1, For Q2: Q2L2)
3. Use meaningful names for each class and the associated variables by following the general naming conventions.
4. For each question, create a static header method containing your name and student number, and a second static footer method with a good-bye message.
5. Comments: Any engineering work has to be well-documented, and that's the norm. It is recommended that you write the comments (it does not have to be a long story) for your code, but it is not mandatory for the Lab-Exercises.
6. Once the assignment is completed, go to your 'Assignments' folder. Select the project folder (e.g. *username_Lab1*). Right-click to select 'Send to' 'Compressed zipped folder'. Make sure it has the same naming convention (e.g. *username_Lab1.zip*). Upload this file on OWL as your submission.

C. Lab Questions

1. Question 1 [5 Marks]

Let's check the execution time of our code:

Write a program to find out the factorial of a reasonably big number (e.g.,30) by using both iterative approach and recursive approach, and display the execution time-difference between these two by using the `System.nanoTime()` method in your code. Use `printf()` method with proper formatting based on your choice (number of places after decimal point, use of scientific notation etc.). Use a very big number for factorial calculation, see what happens; in this case, run the code and make sure that you know why you get infinity. Ask question if that does not make any sense.

`.nanoTime()` method header: `public static long nanoTime()`

Returns the current value of the most precise available system timer, in nanoseconds. This method can only be used to measure elapsed time and is not related to any other notion of system or wall-clock time. The value returned represents nanoseconds (1 Nano second = 1×10^{-9} seconds) since some fixed but arbitrary origin time (perhaps in the future, so values may be negative).

In this question you will create five static methods: The driver method, the header, the footer, a method with iterative solution and a method with recursive solution. You can refer to Unit 1 – part 1, Slide 26,27,49,50 for your information.

2. Question 2 [5 Marks]

Working with arrays and numbers

a) Define a static method called *mma5MethodYourFirstName()* with the following specifications:

- Return type: a double type 1D array reference.
- Formal parameter: an integer type 1D array reference.

The method will perform the following tasks:

- It will accept a 1D array of integer numbers. These integer numbers can be positive, negative, even or odd integer numbers.
- Find the maximum, minimum and average values of all the numbers divisible-by-5 from the array.
- Return a double-type reference of a 1D array that will contain maximum, minimum, and average values of the calculated numbers followed by your student number.
 - For example, if your student number is 251999999, it will return a reference of an array that will contain [maxValue, minValue, averageValue, studentNumber].
- You must make sure that if there is no number divisible-by-5 in the array, your code should generate an average value of 0.00 instead of NAN (NAN is generated when you divide any number by 0).
- From inside the method, print a message before the return statement on how many numbers are there in the array. For example, if your code finds 31 numbers that are divisible-by-5, the message should print:

YourFirstName found 31 numbers that are divisible-by-5.

Note: You are only allowed to use Math.max() and Math.min() methods. No mark will be awarded if you use any method or field (such as Integer.MAX_VALUE) from the Integer Class.

b) Define two other static methods – myHeader() and myFooter() with proper parameters.

c) Define the driver method and do the following:

- i) Call the header method
- ii) Create a 1D array of size N (user will input this size from the keyboard)
- iii) Populate the array with integer numbers (no validation required)
- iv) Call the *mma5MethodYourFirstName()* method.
- v) Print the max, min and the average value.
- vi) Call the footer method.