Lab Exercise 1: Revisiting Java Fundamentals – Part 1

SE2205: Data Structures and Algorithms using Java – Fall 2022

Open Day: September 19, 2022; **Cut off time**: September 24, 2022 @11pm

Prepared 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 **FULL** credit for this lab exercise when you submit the working code. No part-mark will be awarded if the codes do 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. If not mentioned, please 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 must 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 [*nanoTime() is a static method in System Class*] 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. 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 returned value represents nanoseconds (1 Nano second = 1 x 10^{-9} seconds) calculated since some fixed but arbitrarily chosen time (perhaps in the future, so values may be negative).

In this question, you will create five static methods in your class called **TestingExecutionTime**: the driver method, the header, the footer, a method with recursive solution and a method with iterative solution. You can refer to Unit 1 – part 1, Slide 26,27,49 for your information. The method requirements have been outlined below:

(a) Header: Create a public static void header method (with an integer parameter) with the following information (see the sample output):
   Lab Exercise 1 (*this number 1 will be passed as argument when you invoke the header method*)
   Prepared By: YourFirstName YourLastName. (Hardcode it; meaning put the data manually)
   Student Number: YourStudentNumber (Hardcode it)
   Goal of this Exercise: Brief description of the project. (Hardcode it)

(b) Footer: Create a public static void footer method (with an integer parameter) with the following message (see the sample output):
   Completion of Lab Exercise 1 is successful! (*This number 1 will be passed as argument when you invoke the footer method*)
   Signing off - yourFirstName.

(c) The iterative method: Create this public static method based on the appropriate choice of the return type and formal parameter list. This method will return the factorial value when called.

(d) The recursive method: Create this public static method based on the appropriate choice of the return type and formal parameter list. This method will return the factorial value when called.

(e) Driver method: Your driver method will do the following:
   i. Call the header method
   ii. Ask the user to enter an integer number that s/he wants to calculate the factorial for. Assume that the user will enter an integer.
   iii. Validate that the number is either zero or a positive number. No mark will be awarded if the validation is not done with the help of a loop. Ask your instructor if you are not sure about this.
   iv. After activating the System.nanoTime(), create the iterative solution inside the driver, calculate the execution time, and display it on the screen.
   v. Now after activating the System.nanoTime() agian, call the iterative method, calculate the execution time, and display it on the screen. This time will be longer than the time you find in iv and vi for smaller n values. For larger n values, vi will always take more time compared to iv and v. Ask your instructor why that is the case or explore the internet.
   vi. Once again after activating the System.nanoTime(), call the recursive method, calculate the execution time, and display it on the screen.
   vii. Call your footer method.

Sample output (it is just a guideline):

```
======================================================
Lab Exercise 1-Q1
Prepared By: Quazi Rahman
Student Number: 999999999
Goal of this Exercise: Checking the code-execution time!
======================================================
Enter an integer number: 58
Factorial (58) is 2e+78
Time taken by iterative solution inside main: 2.90e-06 sec
Factorial (58) with iterative method is 2e+78
Time taken by iterative method call: 5.30e-06 sec
Factorial (58) with recursive call is 2e+78
Time taken by recursive method call: 6.20e-06 sec
======================================================
Completion of Lab Exercise 1-Q1 is successful!
Signing off - Quazi
======================================================
```

2. Question 2 [5 Marks]

Working with arrays and numbers

In this question, you will create four static methods in your class called **WorkingWithArrays**

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, that is , it will return a reference of an array that will contain [maxValue, minValue, averageValue, YourStudentNumber].
- 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 as outlined in question 1.

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.

<u>Sample output</u> (it is just a guideline) - Run 1

```
=========================================================
Lab Exercise 1-Q2
Prepared By: Quazi Rahman
Student Number: 999999999
Goal of this Exercise: Working with arrays and conditional statements!
=========================================================
Enter array size: 3
Enter value 1: 31
Enter value 2: 79
Enter value 3: -23

Here is the result......

In the array, there was no number divisible by 5
=========================================================
Completion of Lab Exercise 1-Q2 is successful!
Signing off - Quazi
=========================================================
```

<u>Sample output</u> - Run 2

```
=========================================================
Lab Exercise 1-Q2
Prepared By: Quazi Rahman
Student Number: 999999999
Goal of this Exercise: Working with arrays and conditional statements!
=========================================================
Enter array size: 4
Enter value 1: 65
Enter value 2: 33
Enter value 3: -25
Enter value 4: 15

Here is the result......
Found 3 numbers that are divisible by 5
The max is 65.00.
The min is -25.00.
The average of the numbers divisible by 5 is 18.33.
My student number is 999999999.
=========================================================
Completion of Lab Exercise 1-Q2 is successful!
Signing off - Quazi
============================================================
```