

Lab Exercise 2: Revisiting Java Fundamentals – Part 2

SE2205a: Data Structures and Algorithms using Java – Fall 2023

Open Day: September 23, 2023; **Cut off time:** Saturday September 30, 2023 @11.55pm

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's Generic Class (generic class with more than one tag, `ArrayList<E>` etc.) and Exception handling. You need to work on one problem for this lab.

B. Evaluation and Submission Instructions

You will get 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 online within the deadline by carrying out the following instructions:

1. Create a Project with the following name: `username_LabExercise2`
2. For this question create a package (Q_LE2)
3. Use meaningful names for each class and the associated variables by following the general naming conventions.
4. For this question, use the static header and footer methods you created before.
5. Comments: Writing comments for Lab Exercises is not mandatory but it is recommended.
6. Once the assignment is completed, go to your 'Assignments' folder. Select the project folder (e.g., `username_LabExercise2`). Right-click to select 'Send to' 'Compressed zipped folder'. Make sure it has the same naming convention (e.g., `username_LabExercise2.zip`). Upload this file on OWL as your submission.

C. Lab Question

1. [10 Marks]

Working with generics and arrays. Here you will create a data structure that will keep a record of key and values. Here the key will be the year of studies of a group of university students, and the values will be their first names. These students are the leaders in the University Student Council. Here your task would be to find out how many student leaders belong to a specific year.

a) Define a generic class called `Pair <Y, N>` with the following specifications (see the class diagram below): [Hint: this code is given in the class handout]

- Two private data fields: key Y and value N.
- Constructor with both Y and N parameters.
- Getter and setter methods for both the data fields.

Pair <Y, N>
- key: Y - value: N
+ Pair(key: Y, value: N) + setKey (key: Y): void

```

+ setValue (value: N): void
+ getKey():Y
+ getValue(): N

```

b) Define the driver method in a class called *GenericsAndArrays_yourFirstName*, and do the following (Check the class handouts):

- i) Call the header method you created in Lab Exercise 1. This method should contain appropriate info on the current lab Exercise 2.
- ii) Declare an ArrayList type reference variable with Integer-tag and fill out the list with integer values 2, 3, 4, 3, 2, 2 with the aid of Arrays.asList (value1, value2..) method as shown below. These numbers will represent the year of studies of the student leaders. Note that both **ArrayList<E>** and **Arrays** classes are available in **java.util.*** package. ***ArrayList<Integer> anyValidName = new ArrayList(Arrays.asList(2,3,...));***
- iii) Declare a second ArrayList type reference variable with String-tag and fill out the list with String values Harry, Lavender, Ron, Hermione, Luna and Vincent with the aid of Arrays.asList(..) method. These string values represent the names of the student leaders. (FYI: Based on both the Lists, Harry is in 2nd year, Lavender is in 3rd year and so on).
- iv) Create an array of Pair type reference variables, whose size will be one of the sizes of the ArrayLists<> you have created in either (i) or (ii).
- v) Populate this Pair-array by the **key** and the corresponding **value** pairs using the two ArrayList<E> reference variables with the help of the getter methods (Slide 19, Unit 1-P2) from the ArrayList class.
- vi) Now prompt the user to enter an academic year from which the names of the leaders will be printed.
- vii) Validate that the user enters either 2 or 3 or 4. Also validate that the user does not input any non-integer data (Hint. you can take advantage of a try-catch block and catch the exception inside an infinite loop [check Lecture Unit 1, P3])
- viii) Now based on the user's choice, print the names of the leader(s) from that specific year. (Hint: You may use ArrayList<E> to hold the names of the leaders from a specific year, and print the list by printing the ArrayList's name because ArrayList has the overridden toString() method)
- ix) Keep asking the user till she/he does not want to continue.

Sample output:

=====

Lab Exercise 2-Q1

Prepared By: Quazi Rahman

Student Number: 999999999

Goal of this Exercise: fill it out based on your judgement....!

=====

This program prints the names of the student leaders from year 2, 3 and 4.

To see the list of the students from a specific year enter an integer between 2 and 4 when prompted.

Enter Academic Year (2, 3 or 4): 5

Incorrect input! Enter Academic Year (2, 3 or 4): four

Incorrect input! Enter Academic Year (2, 3 or 4): t
Incorrect input! Enter Academic Year (2, 3 or 4): 2
Found 3 leader(s) from year 2.
Here is the list:
[Harry, Luna, Vincent]
Do you wish to continue? (Press y to continue or any other key to terminate): y
Enter Academic Year (2, 3 or 4): four
Incorrect input! Enter Academic Year (2, 3 or 4): 4
Found 1 leader(s) from year 4.
Here is the list:
[Ron]
Do you wish to continue? (Press y to continue or any other key to terminate): n
=====
Completion of Lab Exercise 2-Q1 is successful!
Signing off - Quazi
=====