

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444**Submission date:****Nov 17, 2021**

Workshop 6

Description:

The following workshop lets you practice basic java coding techniques, creating classes, methods, using arrays, Java I/O, inheritance, polymorphism, Exceptional Handling, JavaFx (GUI), Lambda expressions, Functional Interface.

Task 1:

The details for the task are as follows,

- The popularity ranking of baby names from years 2009 to 2018 is downloaded from www.ssa.gov/oact/babynames and stored in files named **babynamesranking2009.txt**, **babynamesranking2010.txt**, . . . , **babynamesranking2018.txt**.
- Each file contains one thousand lines/ records.
- Each line contains a **ranking**, a **boy's name**, **number for the boy's name**, a **girl's name**, and **number for the girl's name**.

For example, the first two lines in the file **babynameranking2010.txt** are as follows:

1. Jacob 21,875 Isabella 22,731
2. Ethan 17,866 Sophia 20,477

So, the boy's name Jacob and girl's name Isabella are ranked #1 and the boy's name Ethan and girl's name Sophia are ranked #2. 21,875 boys are named Jacob and 22,731 girls are named Isabella.

Note: There are some common names for both boys and girls as well.

You have to write a program that asks the user to enter the year, gender, and followed by a name, and displays the ranking of the name for the year. Here is a sample run:

The image displays two screenshots of a Java Swing application titled "Search Name Ranking Application".

The first screenshot shows the input fields:

- Enter the Year: 2010
- Enter the Gender: M
- Enter the Name: Javier

Below the input fields are two buttons: "Submit Query" and "Exit".

The second screenshot shows the result of the search:

Boy name Javier is ranked #190 in 2010 year

Below the result is a prompt: "Do you want to Search for another Name:" with two buttons: "Yes" and "No".

Task 2: (Lambda Practice)

This task asks you to create your own lambda expressions using Functional Interface.

- Define a functional interface named *ArrayProcessor* which has one function *operations* to returns a double and takes a double type array as its argument
- Write a class that defines four *public static final* variables of type *ArrayProcessor* that process an array in the following ways:
 1. find the maximum value in the array
 2. find the minimum value in an array
 3. find the sum of the values in the array
 4. find the average of the values in the array.

- In each case, the value of the variable should be given by a lambda expression.

The class should also define a function

```
public static ArrayProcessor counter(double value ) { ...
```

- This function should return an ArrayProcessor that counts the number of times that *value* occurs in an array.
- The return value should be given as a lambda expression.
- The class should have a *main()* routine that tests your work.
- Ask the user to give array elements.

Workshop Header

/*****

Workshop #

Course:<subject type> - Semester

Last Name:<student last name>

First Name:<student first name>

ID:<student ID>

Section:<section name>

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature

Date:<submission date>

*****/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Document all the classes properly
- Do Not have any debug/ useless code and/ or files in the assignment

Deliverables and Important Notes:

All these deliverables are supposed to be uploaded on the blackboard once done.

- You are supposed to create video/ record voice/ detailed document of your running solution. (50%)
 - Screen Video captured file should state your last name and id, like Ali_123456.mp4 (or whatever the extension of the file is)
 - Detailed document should include screen shots of your output, have your name and id on the top of the file and save the file with your last name and id, like Ali_123456.docx (or whatever the extension of the file is)
- A word/ text file which will reflect on learning of your concepts in this workshop. (30%)
 - Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali_123456.txt

- Submission of working code. (20%)
 - Make sure you follow the “**Code Submission Criteria**” mentioned above.
 - You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.zip**.
- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.

Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any