Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444 **Submission date:** **Date**

Workshop 8

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, Collection Framework, Java Threads.

Task 1:
In this task you will be experiencing on how to perform matrix addition.

- Suppose you have multiple processors, so you can speed up the matrix addition.
- You must implement the following method in parallel.

public static double[][] parallelAddMatrix(**double**[][] a, **double**[][] b)

- Second must implemented method should be a normal sequential process
  public static double[][] sequentialAddMatrix(**double**[][] c, **double**[][] d)

- Write a test program that measures the execution time for adding two 2,000 * 2,000 matrices using the parallel method (by running the multiple threads).
- Also calculate the execution time for adding two 2,000 * 2,000 matrices using the sequential method (calling the method as normal call).

Hint: For parallel addition divide your matrix in to 4 matrices for example the size of 2000 * 2000 can be divided into [0][500], [501][1000],[1001][1500],[1501][2000] then run the addition of all in 4 different threads and at the end join the threads to finish the process one after another and calculate the time of all the threads.
You can use the examples from the slides on how to calculate time for threads (week 9 lecture slides).

Note: Students are supposed to design their own output for the task

Continue to the next page…

Task 2:

- Write a program called ReverseThread.java that creates a thread (let's call it Thread 1).
- Thread 1 creates another thread (Thread 2);
- Thread 2 creates Thread 3; and so on, up to Thread 50.
- Each thread should print "Hello from Thread! <Number of the thread>", but you should structure your program such that the threads print their greetings in reverse order.

Note: Students are supposed to design their own output for the task

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
- Do not have everything in the *main method*.
- Have a separate TestClass with the main method in it.
- Check your inputs if the user is not entering garbage inputs.
- Use exceptional handling or other methods to let the user know if the inputs are incorrect.

Deliverables and Important Notes:

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

- You are supposed to create video/ record voice/ detailed document of your running solution. **(20%)**
  - ○ Screen Video captured file should state your last name and id, like Ali_123456.mp4 (or whatever the extension of the file is) ○ Record voice clip should also include a separate word file with the screen shots of your program's output, state your last name and id, like Ali_123456.mp3 (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 li_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. Also include the instructions on how to run your code. **(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. **(50%)**
  - Make sure your 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.
- For Late submissions, please refer to the late policy for the course.
- 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 source.