COMP 2800: Software Development
Lab Exercise-6

Instructor: Dr. Muhammad Asaduzzaman (masaduzz@uwindsor.ca)
Teaching Assistant: Srivatsan Vasudevan (vasudev4@uwindsor.ca),
Ali Aman (burkia@uwindsor.ca), and Ryan Hermes (hermesa@uwindsor.ca)

Winter, 2024

Lab Duration: Tuesday 4:00-5:20PM, 5:30-6:50PM, 7-8:20PM
Location: Erie Hall 3119
Date: March 4, 2024 **Submission Deadline:** March 10, 11:59PM

Problem Description

In this lab exercise, you will use multithreading in Java to count the number of characters in a text file. There should be a function/constructor that takes the path to a text file and the number of threads as parameters. The program should print the number of whitespace characters after completing the execution.

1. Implement a program that counts the characters in a file without using the thread.
2. Implement the target program using the lock and condition interface.
3. Determine the time required to complete the execution of the two different versions of the program. Use a large file as an input. Add a note.txt file to report the result and explain the difference in your own words.
4. Don't forget to upload your input file and note.txt.

Marking scheme

1. Implement the program without using thread (2 Marks).
2. Implement the program using the lock and condition interface (5 Marks).
3. Report the runtime and explain the differences (3 Marks).

What do you need to do?

- Complete the program.
- Upload the program on the course website. For each lab you will find a folder (Lab-X, X represents the lab exercise number) in the assignment section of the website.
- If you have only one Java file to submit, upload only that file. If you have created multiple files, compress the project folder (zip compression) you created and upload the zip file.
- Rename the file/folder as follows: LastName_FirstName_Lab_LabExerciseNumber (Example: Asaduzzaman_Muhammad_Lab_1.Java or Asaduzzaman_Muhammad_Lab_1.zip).
- Include a plain text file with explanations (such as note.txt) if there are any additional requirements to run your program.
- TA can modify the input to the program.