Assignment 5

Multithreading Programming in Java (10 Points)

Due Date/Time: Thursday, Nov 7th, 11:00 PM.

"No Submission via email. Only Canvas."

Goals: Practice Java Multithreading and openMP with C++ and understand how to solve the race condition issue in both Java and C++ when using OpenMP API.

In this assignment, you will be expected to conduct refactoring on a given Java program that sequentially counts the number of the customers that have balances less than 1000.00$. The customer's information is stored in a text file (accounts.txt) and read by the program into an **ArrayList<Customer> (**of the datatype <Customer>).

Note: The records stored in the file such that each record has 2 lines (fields) ID and Balance.

ID1(Str)
Balance1 (double)
ID2 (Str)
Balance2 (double)
ID3 (Str)
Balance3 (double)
…
…
…

Please download the project with the text file (posted on Piazza and Canvas for your convenience) and do all of the following tasks:

Part1:

(A): **[4 Points]** Rewrite the Java program (refactoring) so that it does the same work in a multithreaded way (counts number of the customers that have balances less than 1000$ in a parallel way).

(B): **[2 Points]** Analyze the two versions' performance (which one is faster in **Milliseconds**) and share the findings.

Part 2:

[4 Points] Write the program in C++ (OOP way) and make it multithreaded using OpenMP and compare its performance against the Java editions.

Note: Your C++ program must first read the customers' information from the same file (accounts.txt) and store them in a vector of customer objects (**vector<Customer>**). Then, the program counts the number of customers that have balances less than 1000.00$.