CS 420
HOMEWORK ASSIGNMENT H2

In the following exercises, "displaying the **elapsed** time" means displaying the time in milliseconds since start of program execution. This means that your program should store the current clock time at the start of the program and then retrieve the current time as required and compute the elapsed time since the time stored at the start of the program. Find the appropriate Java library class for an object that can be used to retrieve the current clock time with adequate precision for your needs.

1. Simple Multi-Threaded programming - 4.

Write a Java application that runs three new threads (identified as `Thread1`, `Thread2` and `Thread3`). Two threads should sleep for 500 milliseconds before displaying the **elapsed** time with their thread identifier on the terminal screen, while the third should sleep for 1300 milliseconds before displaying the **elapsed** time with its thread identifier. Also, make the last statement in your application (the `main`() method) also display the **elapsed** time with a label identifying it with the `main`() method. Note how closely do the sleeping periods match against the reported times and explain your observation. Also explain why the time printed by the `main`() method is located where you observe it.

2. Simple Multi-Threaded programming - 5.

Modify your solution for exercise 1 so that the three threads have different priorities (`Thread.MIN_PRIORITY`, `Thread.MAX_PRIORITY`, and `Thread.NORM_PRIORITY`). Note if this makes any difference in the correspondence of reported and actual times as compared to the version where all threads have the same priority and explain your observation. Print the numeric value of each thread's priority so the user can see what priorities are assigned to the threads.

3. Simple Multi-Threaded programming - 6.

Modify your solution for exercise 1 so that the `main()` method uses `join()` methods to delay the printing of the time at the end of the `main()` method until all its child threads have terminated.

General Instructions:

- Submission is done electronically on canvas.
- Create a zip folder containing three java files (one for each problem above). Write your explanations in a separate WORD file clearly marking the question #. Include the word file in your zip folder.
- Do not print anything. Everything is electronically submitted into one zip folder. Do NOT include .class or any other code file (ONLY .java must be included in your zip)

- All homework is to be completed by each student individually and represent that student's original, unassisted work. Any material copied in any way from other sources must be clearly identified and attributed.

Programming Instructions:

- Put a block of comments at the beginning of every physical file containing program source code that includes your name, the course name and number, information identifying what functions the program is designed to perform, and instructions how to execute the program. (required)

- Each assignment must be submitted on Canvas by the start of class on the day the assignment is due.
- Submit only one zip file with your entire assignment.
- If you have already submitted a homework assignment and then decide you must resubmit the assignment before it is due, you can submit another zip file to replace previous submissions. You can also use comments with the submission to further explain your submission to the grader. You may resubmit as many times as you find necessary before the assignment due date.