

CSCD 467/567 Lab3

Fix the Faulty Solution one to Alternating Threads

Rules: Your code must use Java Language. If your program shows a compilation error, you get a zero credit for this lab assignment. To avoid compatibility issues, I encourage you to upgrade your JRE to latest version of SE 1.7 or 1.8.

Submission: Wrap up all your java files and a ReadMe text file into a single zip file. Name your zip file as *FirstInitialYourLastName*CSCD467Lab3.zip. For example, if your legal name is Will Smith, you should name your zip file as wSmithCSCD467Lab3.zip.

You are required to submit the ReadMe text file along with all your java code. In the ReadMe file you should put your legal full name, description about how to compile and how to run your program. An example of ReadMe file should look like the following:

(This only serve as an Example. Your ReadMe file should contain the similar content.)
Name: Will Smith
Description: unzip the submitted wSmithCSCD467Lab3.zip, you get a folder named smithLab3.
To Compile: cd into folder smithLab3,
 javac *.java
To Run
 java myLa3 4 1 1000000

Before you leave the laboratory, please show the TA or the instructor how your program works, they will give you a score for this Lab assignment.

For archive purpose, please also submit your single zip file on EWU Canvas by following CSCD467-01 Course → Assignments→ Lab3→ Submit Assignment to upload your single zip file.

Problem Description:

You are required to achieve these features in this Lab. **Please use the techniques that we have covered so far. (interrupt and wai()/notify are not an option here.)**

- 1, we have to use two threads T1 and T2 in the main thread.
- 2, starting with thread T1, T1 outputs a message, then thread T2 outputs.
- 3, two threads alternate till each finishes 25 message outputs.

A faulty solution was given and discussed in the last lecture. You are required to work on the faulty solution and fix it in this lab. After you fix the solution, please also answer the following questions about your work.

- 1, Why the provided solution does NOT work? Which statement(s) cause the program to hang up?

2, How did you fix that problem?

3, How does your changes fix the problem? (hints: with regard to the lock)

Correct Program Output

```
Message 1 from Thread T1.  
Message 2 from Thread T2.  
Message 3 from Thread T1.  
Message 4 from Thread T2.  
Message 5 from Thread T1.  
Message 6 from Thread T2.  
Message 7 from Thread T1.  
Message 8 from Thread T2.  
Message 9 from Thread T1.  
Message 10 from Thread T2.  
Message 11 from Thread T1.  
Message 12 from Thread T2.  
Message 13 from Thread T1.  
Message 14 from Thread T2.  
Message 15 from Thread T1.  
Message 16 from Thread T2.  
Message 17 from Thread T1.  
Message 18 from Thread T2.  
Message 19 from Thread T1.  
Message 20 from Thread T2.  
Message 21 from Thread T1.  
Message 22 from Thread T2.  
Message 23 from Thread T1.  
Message 24 from Thread T2.  
Message 25 from Thread T1.  
Message 26 from Thread T2.  
Message 27 from Thread T1.  
Message 28 from Thread T2.  
Message 29 from Thread T1.  
Message 30 from Thread T2.  
Message 31 from Thread T1.  
Message 32 from Thread T2.  
Message 33 from Thread T1.  
Message 34 from Thread T2.  
Message 35 from Thread T1.  
Message 36 from Thread T2.  
Message 37 from Thread T1.  
Message 38 from Thread T2.  
Message 39 from Thread T1.  
Message 40 from Thread T2.  
Message 41 from Thread T1.  
Message 42 from Thread T2.  
Message 43 from Thread T1.  
Message 44 from Thread T2.  
Message 45 from Thread T1.  
Message 46 from Thread T2.  
Message 47 from Thread T1.  
Message 48 from Thread T2.  
Message 49 from Thread T1.  
Message 50 from Thread T2.
```