CSCI5134 Concurrent Programming and Software Modeling

Fall 2015

Programming Assignment No. 4

Due: Sat. Dec 5th, 2015

**Purpose**:

The purpose of this assignment is to understand how to use concurrency in the Java programming language. But as you may already know, Java is an Object Oriented programming language. Therefore, your design must adhere to the OOP methodology to receive credit.

**Requirements**:

In assignment 1, you were asked to modify the producer consumer problem to design a game called "Connect-4" using the C programming language. In this assignment, you are to implement the same problem for playing the game of "Connect 4" except your design must be in the Java programming language, and must follow the object-oriented programming paradigm. Your implementation must use the wait, notify and/or notifyAll facilities to implement the synchronization between threads. Other requirements for how the program is supposed to run are included in the Programming Assignment 1 posted on Blackboard.

**Design Requirements:**

--------------------

Your design must focus on a game class and its contents. For instance a game will have a board with the ability to check for wins and modify itself based on a player's input. A referee should be able to check that board for wins or ties by utilizing the board's check for wins/ties capabilities, this is also part of the game. Also the game will have two players taking turns or not taking turns (threads) playing the game as in assignment 1.

You may include any classes in addition to the one's mentioned above as you see fit. But you must at least have the classes mentioned above and show how they're working together to build a game.

Your program must run exactly as assignment 1 was supposed to run. The new design is only implementation details, but the execution of the code must be identical between this assignment and assignment 1.

Extra Credit: 15 % of the assignment grade.

------------

To receive the extra credit, you need to provide the correct related UML diagrams for your design. You may even have them generate code for you to use in your development. In order to receive credit, you must utilize at least three of the UML diagrams discussed in class, and they must adhere to your implementation as described above. The UML diagrams must represent the entire application.

Your programs should follow good software engineering practices, i.e., **all constructors should call the fully parameterized constructor, constructors in the subclass should call the superclass constructor appropriate to initialize the superclass state, use of get and set methods, overriding methods should call the superclass version of the method to do partial work and OF COURSE use of private fields AND ONLY private fields in your classes**

Zip your project folder and upload it to blackboard on the due dates before midnight and schedule a demo with TA the week after the assigned due date.