CS110 Assignment #10
Due Wednesday, April 29th (no late assignments will be accepted)

You have over two weeks to complete this assignment. It is not trivial, do not delay. Read through the requirements and grading rubric provided – don't lose points for missing a requirement.

If you achieve a higher grade on this assignment than your lowest homework grade, I will count this assignment twice, once as assignment #10 and once as a replacement for your lowest homework grade.

When seeking help from the TAs or Jackie, your code must be clean –proper indentation/formatting with at least some short comments. Programs are getting too complex for us to be wading through unformatted, undocumented code. When coming for help, come with specific questions. "It doesn't work.", "It keeps crashing" aren't helpful. "I created a deck of cards, and ensured that all 52 cards were created. When I deal the cards to 2 hands, I find a duplicate card" is much more helpful.

There will be NO LAB ON WEDNESDAY, APRIL 29TH

For this last homework assignment you will be creating a GUI-based implementation of the kid's card game, War. If you are unfamiliar with the game, you can learn about the game here http://en.wikipedia.org/wiki/War_%28card_game%29 There are some variations in the way the game is played. For our implementation, when a "war" occurs

- Both players play the next one card of their pile, face down, and then another face up
- If a player runs out of cards during a war, they will immediately lose

Design decisions are yours to make; you will be graded on these design decisions. The only requirement is that you must use inheritance at least once (outside of the GUI implementation). I would encourage you to sit down with a deck of cards and either play a game with a friend, or simulate a game yourself.

I have provided a zip file with the card images that I used. You may use these or provide your own.

Submission

- You will submit using GitHub (submit a link to your repository to Blackboard by due date). If you missed this lab, you can find a handout describing the process on Blackboard.
- In addition to any java files and image files, submit a .html file for each class containing the javadocs documentation (in jGrasp, File->Generate Documentation).
- Note: nothing will prevent you from altering your files in your GitHub repository after the due date; however, all updates have a time stamp associated with the files. The last update prior to the due date/time will be graded.

Grading

Design decisions 25
- appropriate class choices
- use of inheritance/polymorphism
- clean, clear public interface to each class

Implementation 40
- Correctness
- Reasonably efficient, well-implemented
- Separation of game implementation and GUI

GUI 20
- Clean, intuitive design and implementation
- Appropriate choice of components

Documentation 15

- Complete documentation
- javaDocs files submitted

GitHub 10

- Repository complete with all files
- A history of prior versions