CS162
Assignment 6 (Extra Credit), Graphical Guess My Number

For this extra credit assignment you will create a graphical version of a guess my number program.

Be sure to check out the previous assignment solutions on command-line arguments, File IO, GUIs, and any other related materials.

Your program needs to have the following **behaviors** (more is okay, but make sure they do not interfere with using these functionalities directly):

1. Take a filename as a command-line argument
2. Read two numbers from a file where:
   a. each one is on a different line
   b. the first number is for the minimum value able to be generated for guessing
   c. the second number is for the maximum value able to be generated for guessing
3. Generate a number in the range from the first number to the second number
4. Graphically display some message to the user stating the range of possible numbers
   (this can just be a label or text box)
5. Have a place in the GUI to type or select a guess
   (this can be a textbox, drop-down list, scroll-list, or maybe even a number of buttons equal to the range of numbers possible to guess)
6. Display whether the number guessed is higher, lower, or correct
7. Somehow show the new range of possible numbers based on the recent guess

Your program needs to be setup in a certain fashion to make parts of it easy to test, you should have the following **methods** (more are okay, but make sure they do not interfere with using these methods directly):

1. int getMinNum(), takes no arguments and returns the minimum value able to be generated
2. int getMaxNum(), takes no arguments and returns the maximum value able to be generated
3. int generateNumber(int, int), takes two ints, first is bottom of the range and second is the top of the range, returns a random number in the range provided
4. int guess(int), takes one int that is the guess and returns a 0 for a correct guess, a -1 to indicate that a guess is too low, and a 1 to indicate that a guess is too high
5. int[] getCurrentRange(), takes no arguments and returns an int array of size 2, where the first array element is the bottom of the current range available and the second array element is the top of the current range available
   (this available range is the range that is modified when a user guesses a number, it is used to let the user know what the highest and lowest values they **should** guess are, based on the initial range and previous guesses)
   For example, when an initial range is from 4 to 12 and the random number happens to be 9, if the user incorrectly guesses 6, then the program should say that the number is higher, and the new range that the user should guess in should now be 7 to 12.

Your program should consist of the following three **classes** (more are okay, but make sure they do not interfere with using these classes directly):

1. one for the program logic called Guess
   (should hold the random number, range, and all of the required methods from the method requirements above)
2. one for the GUI called GuessGUI
   (should just be the GUI and the methods required to get and set values from the GUI)
3. and one for your main called GuessDriver
   (should setup a Guess object, GuessGUI object, then have a control loop for getting values

from the GuessGUI and use the Guess object to determine what values to set back into the GUI)

The assignment will be graded based on the following (and points possible in parens):

1. (15) Does the program look like a guess my number program, compile, and run
2. (25) Does the program meet the 7 **behavior** requirements (checked briefly by inspection, so make it obvious what each component does)
3. (25) Does the program have at least the 5 **method** requirements, with the signature and behavior as requested (tested using unit tests, feel free to share your own unit tests with each other on the discussion boards if you would like)
4. (15) Does the program have the 3 **class** requirements (checked similarly to the methods)
5. (5) is there encapsulation of objects (remember to use private variables and getters and setters)
6. (5) is the code well documented
7. (5) is there any error handling (file not found, a guess other than a number)
8. (5) is there a design document showing how the program should run (try using a UML class diagram)

Be sure to turn in your files to TEACH by 16 September, 2012 by 23:59. **I will not be accepting late assignments on this one!**