**Final Assignment**

Final Assignment:

Due Date: November 4, 2013.

Implement one of the following.  In all cases, be sure that the user interface is graphical; no text-based user interfaces.  Make use of Java’s object-oriented features, such as classes and inheritance.

Note that for this assignment, you will have to sign up for a date/time to both demo your code, and to perform a code review with a group of senior level Java developers.  You will do both at the same review session.  Prepare to dazzle them, and to receive some extremely valuable feedback on your coding style, including tips and an appraisal of your code.  Stay tuned for information about how to submit your assignment:

1. Assignment Possibility 1: Create a simple loan amortization program:
   1. Allow users to specify the origination date of the loan.
   2. Allow users to specify the amount of the loan.
   3. Allow users to specify an interest rate.
   4. Allow users to specify the loan term, in terms of years (e.g. - 30 years)
   5. Allow users to create a list of "pre-pay" instances.  That is, dates on which the user will make additional payments toward principle.  Prompt the user to specify the date and the amount of each of the prepayments.
   6. When users click a "Calculate" button, display a window that includes these data elements:
      1. Payoff date.
      2. A graphical table (I recommend using the JTable class) that lists the following data for each payment:
         1. Monthly payment amount.
         2. Due date.
         3. Additional payments toward principle.
         4. Outstanding loan balance on the due date for the payment.
      3. Make each window that the “Calculate” button displays a free-standing window of its own (I recommend using JFrame) so that users can put loan payoff windows next to each other to compare them.
2. Assignment Possibility 2:
   1. Create a horse-racing game.
   2. This game will allow users to maintain an account with a certain balance of funds.  The user will be allowed to bet on races to win or lose money.
   3. Don’t worry about allowing multiple users or user login functionality; assume that the user is the one and only user that will ever use this game.  You don’t have to provide login or logoff capability, nor must you maintain information for multiple users.
   4. When the game starts, read a text file to determine the latest information about the user, which should include their name and the amount of money in their account, in USD.
   5. Provide the user with the ability to visit these race tracks:
      1. Lonestar Park
      2. Santa Anita
      3. Gulf Stream
      4. Churchill Downs
      5. Bellmont
   6. Each track should have an associated list of races going on, the information of which should exist within a file in the file system.  Each race should have an associated list of horses running in the race.  Allow the user to select the track and the race on which they’d like to bet.  Only allow them to bet on races that have not yet already happened, and that are next in line to be run; don’t allow the user to bet on a future race.
   7. Once the user has selected a track and a race, show them a list of horses running in the race, including the horses’ names, numbers, and odds.  Allow them to select a horse, a bet amount, and whether they’d like the horse to win, place or show.  Don’t worry about more exotic bets [unless you want to have additional fun with this assignment].
   8. After the user has selected their horse, make a button labeled “Post Time!” available.  When the user clicks this button, start the race.
   9. Randomly determine the place of each horse in the race, based on the horse’s odds.  Add a fudge factor to make the results of the race less predictable.  If you need help with this one, then please give me a shout and I’ll talk you through it.  It’s not as hard as it sounds.
   10. During the race, display the progress of the race.  Be sure that each race takes at least thirty seconds to run, in order to get the player’s adrenaline going.  Get creative with the way you display the progress of the race.  Maybe represent each horse with a small horse icon?  Maybe with just an X?  I’ll let you decide.
   11. When the race is over, display the results, and display the player’s winnings/losings, the player’s old account balance, and their new balance.
3. Assignment Possibility #3: Blow the dust off your sliding tile puzzle game.  Make it graphical using Java Swing.
4. Assignment Possibility #4: Other. . .anything you like.  Be sure that you use Java Swing and that you make use of object-oriented features within Java.