Horse Racing Assignment

# Assignment Three – ICS 3U AP

The following assignment is meant to allow you to test what you have learned to date in a practical manner by creating a text (console-based) game – Horse Racing.

The topics covered in this assignment are:

**Current Unit - Six:**

1. Using Arrays
   1. (String array – used to hold the list of horses)
   2. (String array – used to hold the list of players)
   3. (Integer array – used to hold the amount a player’s wallet)
   4. (Integer array – used to hold the amount a player has bet)
   5. (Integer array – used to hold the horse a player has bet on - #)

**From a Previous Unit - Four and Five:**

1. Selection (if statements)
2. Repetition (while and for loops)
3. Boolean Expressions
4. Error checking and validating input (try … catch, parsing data)
5. Wrapper Classes (Integer, Double – using parseXXX methods)
6. Reading from files (new File() – covered in programming club initially)

**From a previous Unit - One to Three:**

1. Variables, Constants
2. Proper coding conventions
3. Writing Comments
4. Assignment Operator
5. Strings and primitive data types
6. Random Numbers

In attempt to ensure that this assignment (or game) is fun to play while at the same time manageable to design and implement there are some areas that will require some creative problem solving skills.

# The Assignment

In this version of horse racing a list of horses will reside in a file with the following format:

Number of Horses to Follow – type int

List of Horse Names – type String

The file that you may use can be obtained from Blackboard and contains a list of 86 horses – famous horses. The file is customizable. The user can add, remove and change the names of the horses.

All races will consist of anywhere from 5 to 8 horses that will be randomly picked from the String Array that you will load from the file. Think about how you can randomly choose an element (horse) from the list.

Also, in this game you will be able to have multiple betters that when the game begins will also be loaded in from a file with the same format as the horse data. The different between this file and the horse data file is that the player file will also contain the amount of money in the player’s wallet. You can assume that each player’s name consist of a single word.

Example of the playerData.dat file:

**5**

**Jack 500**

**Steve 1000**

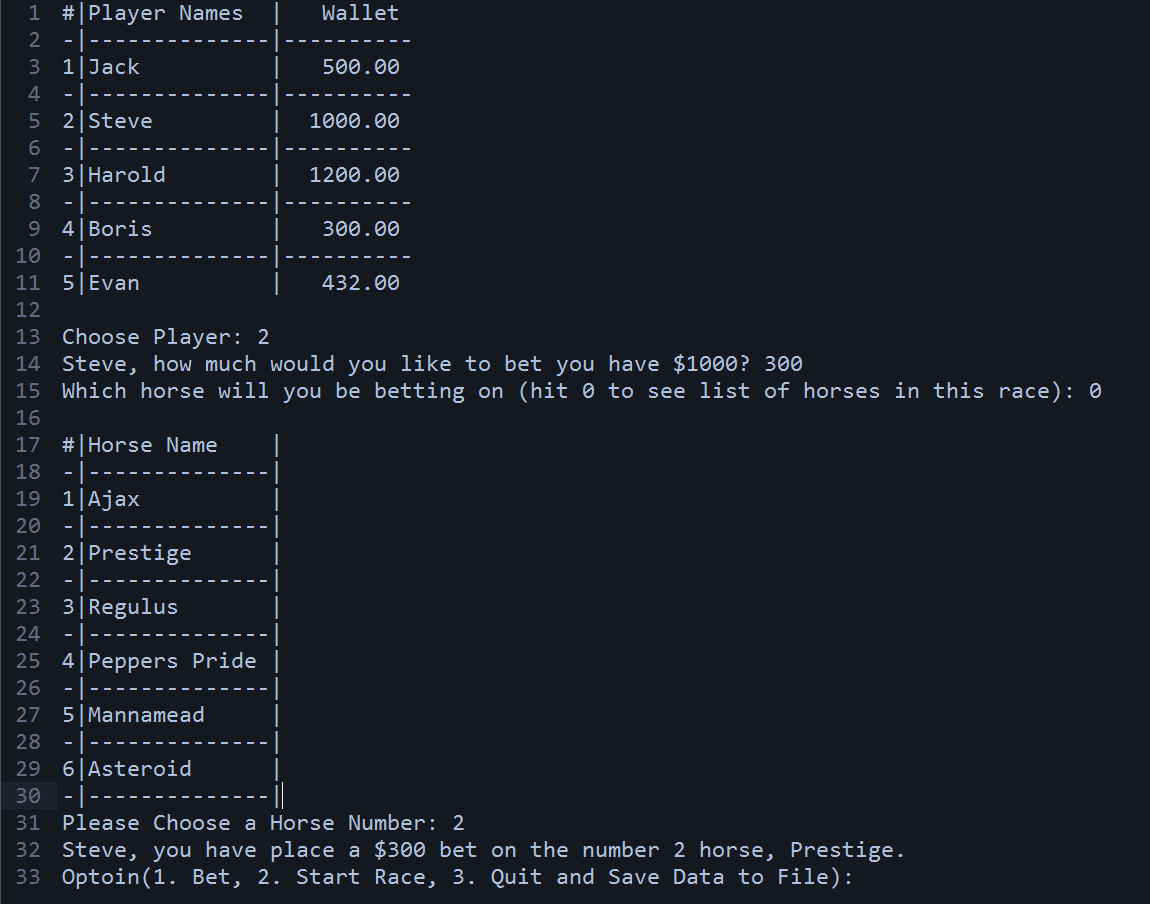
**Harold 1200**

**Boris 300**

**Evan 432**

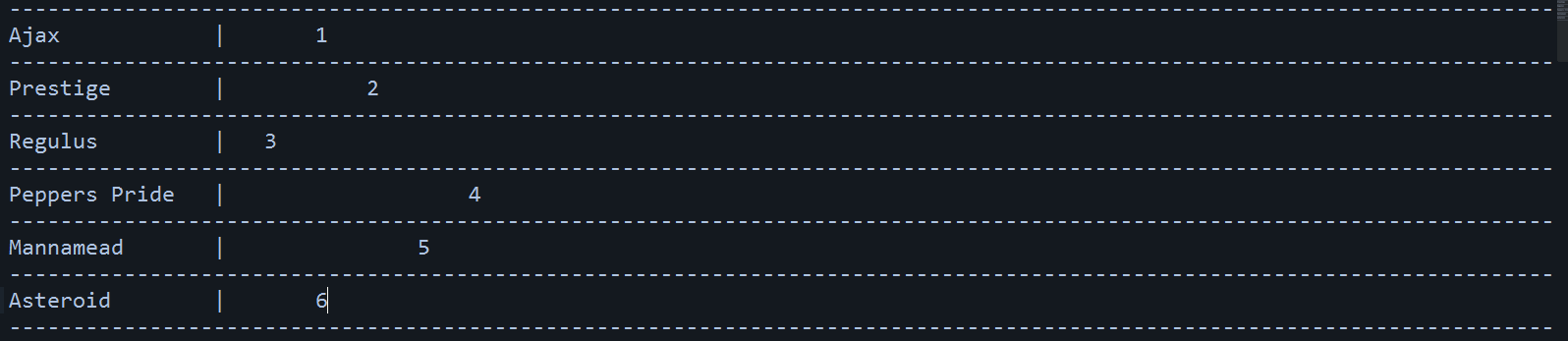
At the start of each race, the users will be presented with the horses that are currently in the race. Each user will have the opportunity to place a bet or choose to not bet. How you do this is up to you; however, if I were doing this I would have a prompt under the list of horses in the race that gives the option to place a bet or start the race.

If the user selects place a bet, they will be displayed a list of the players with the amount of money in their wallet. The user will select a player by number (validation will occur) and then they can place their bet for that player (more validation will occur.



You will notice that there is an option to save the data to a file, when you quit the game it will overwrite the player data file with the updated player data. The file should have the same format as it was read.

Once a race starts, the track will look as follows:



You will notice that the positioning of the horse is indicated by where its number is on the track. The race will consist of a series of time-steps that can be controlled. After each time step the program will redraw the track with new positions of the horses. This will continue until one horse reaches the end.

In this version of horse racing, there are no odds, and you can only bet on the winner of the race. That means you cannot bet a horse to place or show, box them, create a trifecta or exactor.

In addition, to what I have outlined above, you will need to add one extra feature to this game. Hopefully, everyone adds a different feature but there is no restriction that you cannot have the same feature as someone else.

# You will need to hand in the following when you submit your solution.

1. Your code – as a zip file containing the Java Project – not just the .class file
2. Write up that includes:
   1. Algorithm of the solution (Using a flow charting tool)
   2. Instructions on how to play your game with screen captures of the actual game
   3. A cover page
   4. A table of contents (generated using the correct headings from Microsoft word)
   5. A Section containing what you liked about the assignment
   6. A Section of what you found difficult/challenging about the assignment
   7. A Section about what you would change about the assignment

I would suggest using an application like Lucid Chart (A Google App – that works out of Chrome to create an charts for your technical courses.

**Due Date: Friday January 30th, 2013**