**Assigment #1 - Creating Objects**

**Assignment Instructions:**

**20% of your final grade**

This assignment reflects one of the most common scenarios in bioinformatics: a  
series of files that are linked with each other based on (different) identifier numbers.

It also asks you to start thinking about "databases", and the fact that databases  
are dynamic and must be updated (in this case, the amount of seed left in your genebank)

THIS ASSIGNMENT WILL PROBABLY TAKE 8-10+ HOURS TO COMPLETE - do not delay starting it!  
I am giving you 2 weeks to complete the task (Tuesday at midnight).

Submit by committing your code to GitHub:

1. Create a new project "Assignment Answers"

2. Create a new folder "Assignment 1"

3. Deposit your files into that folder.  Commit/push them when you are finished

4. Email me your GitHub username so that I can retrieve them.

Also, do not be worried about asking me questions - I will answer as quickly  
as possible! \*\*I WILL SEND BOTH THE QUESTION, AND MY ANSWER, TO ALL CLASS MEMBERS\*\*  
so that you do not have an advantage; however, I will send this information anonymously.  
The person asking the question will not be revealed.

==================================

**ASSIGNMENT:**

There are three tab-delimited data files:  
1. seed\_stock\_data.tsv  
2. gene\_information.tsv  
3. cross\_data.tsv  
  
#1 contains information about seeds in your genebank  
#2 contains information about genes  
#3 contains information about the crosses you have made

Each file begins with a heading line, followed by lines of data

**\*\*\*\*\*\*\*\*\*\*\*\* YOUR TASKS \*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Your task is to use Object-oriented programming to achieve two things:**

**1)** "simulate" planting 7 grams of seeds from each of the records in the seed stock genebank  
then you should update the genebank information to show the new quantity of seeds  
that remain after a planting. The new state of the genebank  
should be printed to a new file, using exactly the same format as the  
original file seed\_stock\_data.tsv

-- if the amount of seed is reduced to zero or less than zero, then  
a friendly warning message should appear on the screen. The amount  
of seed left in the gene bank is, of course, not LESS than zero

**2)** process the information in cross\_data.tsv and determine which genes are  
genetically-linked. To achieve this, you will have to do a Chi-square test  
on the F2 cross data. If you discover genes that are linked, this information  
should be added as a property of each of the genes (they are both linked to each  
other).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Hints:**

\*\*\* You will need to create 3 Objects:  
1. A Class for the Gene  
2. A Class for the SeedStock  
3. A Class for the Hybrid Cross  
  
\*\*\* the values of some Object Properties will be other Objects

\*\*\* the seed\_stock ID is the key to link seed\_stock\_data->cross\_data

\*\*\* the GeneID is the key to link gene\_information->seed\_stock\_data

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

**The output of your program should look like this:**

$ ruby process\_database.rb gene\_information.tsv seed\_stock\_data.tsv cross\_data.tsv new\_stock\_file.tsv  
WARNING: we have run out of Seed Stock B52  
WARNING: we have run out of Seed Stock A51  
Recording: ufo is genetically linked to pi with chisquare score 32.2794279427943

Final Report:

ufo is linked to pi  
pi is linked to ufo

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

**Your program should also create the tab-delimited updated new\_stock\_file.tsv that contains:**

Seed\_Stock Mutant\_Gene\_ID Last\_Planted Storage Grams\_Remaining  
B52 AT5G20240 8/2/2015 cama16 0  
A348 AT4G36920 8/2/2015 cama25 5  
B3334 AT3G54340 8/2/2015 cama18 15  
A334 AT1G69120 8/2/2015 cama2 21  
A51 AT1G30950 8/2/2015 cama25 0

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

**\*\*\*\*\*\*\*\*\*\*\*\*\* BONUS \*\*\*\*\*\*\*\*\*\*\*\*\*\***

BONUS SCORES   
+1% if your Gene Object tests the format of the Gene Identifier and rejects incorrect formats without crashing

* Arabidopsis gene identifiers have the format /A[Tt]\d[Gg]\d\d\d\d\d/
* If the identifier isn't correct, then your code should stop with a helpful error message

+1% if you create an Object that represents your entire Seed Stock "database"

* the object should have a #load\_from\_file($seed\_stock\_data.tsv)
* the object should access individual SeedStock objects based on their ID (e.g. StockDatabase.get\_seed\_stock('A334')
* the object should have a #write\_database('new\_stock\_file.tsv')

To get these extra scores, your code needs to demonstrate that the functions really work!  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*