Assignment #3

1. In at least one of your data classes from the previous assignment, add another field of type
   1. enum.
   2. i.e. If one of your data classes was Teacher for example, add a field called Gender which is an enum and could be MALE or FEMALE
   3. keep all your enums as a separate class file and put them in the model package
2. Modify your util class(es) to create 100 Random instances with java.util.Random, also populating this new enum field you just created above.
3. Create a relationship between your 2 data classes
   1. i.e., a Teacher can have multiple Students. And Students can have multiple Students
   2. create a fictitious use case (now this is very, very, very specific to your project /idea) that deals with the relationship between these data classes
   3. In your util class (or another util class), create a Report that deals with this relationship
   4. Create a fictitious use case where 5% of the time something bad happens in the printing or populating (of these 100 random instances). When this “bad” thing happens,
      1. a RuntimeException (i.e. IHateThisTeacherException – say it happens when teacher’s name is XXX) should be thrown.
   5. Create a fictitious use case where 1% of the time something very, very bad happens in the printing or populating (of these 100 random instances). When this “bad” thing happens,
      1. Create an Exception (i.e. IllegalImmigrantException, i.e. when your Teacher has the lastName: FromPlanetAzeroth).
   6. Create a separate package for your exceptions
4. Update your Test Driver, reporting util classes as necessary.

* When an exception occurs, you should “catch” it, print it out and continue with the rest of your Test cases

Also begin documenting your changes (changes.html) on what has changed from your previous release

Hints: A lot of this assignment is very specific to your project. If you want to redo assignment #1 and #2 from scratch, be my guest. Part of the purpose of this assignment is for you to show me you understand enums and exceptions (checked vs unchecked, printing stacktraces) and get you more familiar with other java.util classes like Random. If writing a new proposal makes more sense to you, then be my guest and start your project from scratch

The other part of the purpose of the assignment is to show me you know how to design your testcases and data models (IMHO, this is more important)

One more rule on marking and Github. I’ll be marking you’re your assignment off of the “master” branch. Let’s say I checkout your code say 3 days after the previous assignment, but 4 days before you next assignment. What do you think the chances are that I’ll pick up some of you upcoming changes for the next assignment or your Work In Progress if you check-in on a regular basis? Bottom line: I will always be looking at the “master” branch. If you want to create another branch on Git, that is actually recommended. To be more exact, this isn’t a problem specific to this lab or course, but a general problem (or rather fact of software development) … how does one deal with a “golden” release and how do other developers keep on developing when your previous release has been stamped? The answer: “branches”