Assignment 3: Basic Java coding and JUnit

This assignment assesses your basic knowledge of Java and JUnit, which you will need for future assignments and projects.

**To complete the assignment you must complete the following tasks:**

1. Clone your individual GitHub repository in your local workspace. This is the same repository that you used for the previous assignment (i.e., https://github.com/gt-ud-softeng/6300Fall15<your GT login name>).
2. Download the archive [Assignment3.zip](http://measure.cc.gt.atl.ga.us/classes/seclass/Assignment3.zip)
3. Unzip the archive in the root directory of the repository, which will create a directory called Assignment3 and several subdirectories. Hereafter, we will refer to the directory Assignment3 in your local repo as <dir>.
4. Directory <dir>/src contains, in a suitable directory, Java interface edu.gatech.seclass.MyStringInterface. It also contains exception edu.gatech.seclass.IllegalIndexException, which is used by the interface.
5. Your **first task** is to develop a Java class called MyString that suitably implements the MyStringInterface that we provided. (The semantics of the methods in the interface should be obvious from their name and from the comments. If not, please ask on Piazza.) Class MyString should be in the same package as the interface and should also be saved under <dir>/src/edu/gatech/seclass.
6. Your **second task** is to develop a set of JUnit 4.0 test cases for class MyString. Specifically, you should create four test cases for each method that implements a method in the interface, except for getters and setters (i.e., the first two methods). Make sure that every test method has a suitable oracle (i.e., either an assertion or an expected exception), that the tests are not trivial, and that at least one of the tests for a method that can throw an exception results in an expected exception. In other words, each test should (1) test a specific piece of functionality and (2) check that such piece of functionality behaves as expected. To make your job a little easier, the archive also contains, in directory <dir>/test, a test class edu.gatech.seclass.MyStringTest, which provides a skeleton for the test cases and a complete implementation for a few of them. Your job is to write the body of the test cases that are not implemented. Make also sure that the test cases you write are not just a trivial variation of the ones we provide. Finally, creating more than the required test cases will not be rewarded nor penalized; in other words, feel free to write more tests if you are so inclined.
7. Submit your solution by
   * Pushing <dir> and all the files and directories underneath it (except for the files excluded by .gitignore, if you have one, such as all class files) to the individual remote GitHub repository we assigned to you (https://github.com/gt-ud-softeng/6300Fall15<your GT login name>, as usual). In particular, the repo should contain files
     + Assignment3/src/edu/gatech/seclass/MyString.java
     + Assignment3/test/edu/gatech/seclass/MyStringTest.java
   * Submitting the commit ID for your submission on T-Square. You can get your commit ID by running "git log -1" and copying the first line of its output, which should be in the format "commit <commit ID>"

**Notes:**

* You cannot modify the provided interface (MyStringInterface), nor the already provided test cases, except for the ones that you are supposed to implement, obviously, whose body is simply "fail("Not yet implemented")".
* We will also run your code against our set of test cases to make sure that you implemented the functionality of the required methods correctly. Note that our test cases are fairly simple and are not trying to exercise arcane corner cases, so as long as you implement the interface as described, you will be fine.
* Although it is not mandatory, we recommend that you use Eclipse to complete the assignment, so that you can use the assignment to also get familiar with this IDE (in case you are not already familiar with it). If you do so, feel free to commit Eclipse related files (e.g., .classpath and .project) to the repo as well. Please note that we provide such files in the archive, so you should be able to import Assignment3 into Eclipse as an existing project by doing the following:
  + In Eclipse, select File->Import->General->Existing Projects into Workspace
  + Click “Next”
  + Select <dir> as root directory
  + Make sure that (1) “Assignment3” is selected under “Projects:” and (2) “Copy projects into workspace” is not selected
  + Click “Finish”
  + Assignment3 should at this point be a Java project configured to use Java 1.7 (you will have to change this in the project properties if you do not have Java 1.7 installed; in this case, you should see an exclamation mark next to the project).
  + You can commit and push directly from Eclipse or from the command line
* Please also note that using Eclipse (or another IDE) would also make it much easier to create skeleton code for the class that implements the interface, create and run the JUnit test cases, and so on.
* Before submitting, make sure to compile and run your test cases and to check that they all pass--something else that you can do at the push of a button within Eclipse.
* You can perform multiple commits as you produce your solution. This is not only fine, but actually very much encourage.
* We also encourage you to use a “development” branch and merge your stable version(s) into the “master” branch. If interested, see [this site](http://nvie.com/posts/a-successful-git-branching-model/) for an example of a possible branching model of this kind.