Practice Timed Lab

1 Introduction

This practice timed lab will prepare you for the actual timed labs you will see in future weeks. You will have 30 minutes to complete the assignment.

2 Using the Submission Tool

Included with this timed lab is a handy-dandy submission tool. You will not submit to T-Square, but will use this instead. The commands are:

- `ant` or `ant compile` compiles your code. Basically just like running `javac PracticeTL.java`, but you don't need to specify the java file. Just run `ant compile`
- `ant test` runs the provided JUnit tests. For this practice timed lab, the output will perfectly correspond to your hypothetical grade. For actual timed labs, output will roughly correspond - you can use this to see if you're doing something wrong, and get an idea of how well you're doing.
- `ant checkstyle` is a convenient way of running checkstyle without typing that obnoxious command.
- `ant run` will run the code.
- `ant submit` will actually submit your code to the GT Github repo. This will prompt you for your user name and password each time, but you can run it as many times as you want if you want to keep resubmitting as you make minor changes.

3 CIA Redactor

You will be building a redactor for the CIA. We'll walk you through the process. You will need to write several helper methods, please do so exactly as directed.

3.1

First, write a method `public static String generateBlackout(int length)` that generates a string composed of the the $*$ character that is $length$ long. For example, if I call `generateBlackout(5)`, I should recieve "*****" back.

hint: use a for loop

3.2

Next, write a method `public static String redact(String sentence, String[] blacklist)` that replaces any occurences of strings in the blacklist with a redaction string of the same length. That's a mouthful, so for example, if "treadstone" was in the blacklist, and *sentence* was "Jason Bourne was a key operative in the treadstone program", then the method would return "Jason Bourne was a key operative in the ********* program"

You do **not** need to worry about case sensitivity. Just replace exactly the strings in the blacklist.

hint: use the method you've already written, a for loop to loop over everything in the blacklist, and check the `String` API for a method that would simplify replacement.
hint: Remember that strings are immutable! Methods you call on strings return new strings, they don't change the old ones.

3.3 Putting it together

Finally, we've provided the method `public static String[] redactDocuments(String[] sentences, String[] blacklist)` for you. Using the methods you've written already, transform the sentences array into a redacted array. That is, create a new array of the same length, and for every item in the original array, add the redacted form of that item to the new array.

We've provided an example test in the `main` method to get a sense for what's being asked of you, in addition to the tests that you can run.

4 Checkstyle

You must run checkstyle on your submission. The checkstyle cap for this assignment is **0** points.
Review the Style Guide and download the Checkstyle jar. Run Checkstyle on your code like so:

```
$ java -jar checkstyle-6.2.1.jar *.java
Audit done. Errors (potential points off):
0
```

The message above means there were no Checkstyle errors. If you had any errors, they would show up above this message, and the number at the end would be the points we would take off.

The Java source files we provide contain no Checkstyle errors. For this assignment, there will be a maximum of **0** points lost due to Checkstyle errors (1 point per error). In future homeworks we will be increasing this cap, so get into the habit of fixing these style errors early!

5 Without Ant

1. Compile, run, and use checkstyle as normal.
2. Submit with `java -jar lib/Submit.jar PracticeTL.java`

6 Confirm Your Submission

Your submission was pushed to a git repository on github.gatech.edu, which should open in a new browser window after you run `ant submit`.