Timed Lab 3
Collections, Streams, and Lambdas

The Magnificent Evan Cahill & Thomas Shields

1 Problem Description

This is a two part timed lab. The first part will cover the collections API. The second part will test you on your knowledge of the stream API and lambda expressions. Each file will be graded separately from the other files, so be sure to attempt all the parts. Please see the instructions for each of the parts below:

Part 1: Collections (100 points)

This part of the timed lab will require that you modify one file and create another:

`Assignment.java`

1. The `Assignment` class represents an assignment in CS 1331 and is composed of the `title`, `category`, and `maxGrade` of the assignment.
2. Instances of the `Assignment` class must behave properly as elements of any collection. For example, equivalent `Assignment` instances should be findable via the contains method of any collection class and duplicates should be removed in any Set class.
3. The significant fields of an `Assignment` are its `title` and `category`.
4. Lists of `Assignment` instances should be sortable using the `public static <T extends Comparable<? super T> sort(List<T> list)` method according to their natural order. Their natural ordering orders them first by their `category` and then by their `title`. *Hint: All enums have a natural ordering , which orders them according to the order that they were declared in the enum declaration.*

`GradebookAssignmentComparator.java`
should be a strongly typed Comparator that compares only `Assignment`s based on the maxGrade field

`Category`

1. Is provided for you
2. The `Category` enum represents the category of an `Assignment`.
3. The enum has the following values `HOMEWORK`, `EXAM`, and `LAB` declared in that order.
4. You do **NOT NEED** to modify this.

We have provided a main method in `Assignment.java` for you to test you solution. You can use it with `ant run`, but if you want to run it manually, you will need to use the -ea switch when you run your program to enable the assertions. ex.

```
java -ea Assignment
```

Part 2: Streams & Lambdas (Up to 20 Extra Points)

NOTE: This is an extra credit portion. "ant test" will only say if you did it correctly, otherwise you may assume you have done it incorrectly. We will need to grade part of this manually, however, so for this section the ant test command may be slightly inaccurate.

`StreamEncoder.java`

You have been provided with a `StreamEncoder` class which contains several methods for working with encoded Strings. Your job is to complete the `decodeSentence(String)` method which takes in a sentence and returns the decoded sentence. You must use stream operations, such as map and reduce, and lambda / method references in your solution. To help you with this task you have been provided two helper methods:

1. `Stream<String> getWordStream(String)`
   Splits a sentence into words based on whitespace and returns a `Stream<String>` over the words in the sentence.
2. `String decodeWord(String)`
   Takes a single encoded word and returns the decoded word. Do not worry about how this method works, but if you are interested it uses ROT47 encoding.

Once again we have provided a main method that will print out a secret message when you have correctly implemented the `decodeSentence(String) method.`

2 Solution and Tips

We've provided all the source files you will need, but you will need to modify them. **Do not change any file names or things marked as do not modify.**

It is absolutely vital that your submission compiles. Non-compiling solutions will receive a zero: no exceptions!

Make sure to test your code.

3 Checkstyle

10 point cap. Run using ant: "ant checkstyle"

4 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 Java source files.
- `ant test` runs the provided JUnit tests. For this timed lab, the output will be your grade. Of course, we reserve the right to make changes to the grading rubric if necessary, but if you pass these tests, you've done it correctly.
- `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.

5 Confirm Your Submission

Your submission was pushed to a git repository on github.gatech.edu. After running `ant submit`, a new browser window should open at the repository, at https://github.gatech.edu/[your-username]/[your-username]-timedlab3