CPSC250L Lab 6
Text I/O and `String` Formatting

Spring 2018

1 Introduction

The focus for this lab is file input and output as well as string formatting. Check the JavaDoc for `File`, `Scanner`, `PrintWriter`, and `String.format`.

2 Exercises

2.1 Line Numbers

In this exercise, you will read in a file, prepend line numbers to each line, and then output it to another file.

Exercise 1

Create a class named `LineNumbers` and implement the following method in it.

- `public static void process(File input, File output)`

  This method receives a text file. The method writes all lines from the input file to the output file with each line preceded by its line number formatted as follows:

© Christopher Newport University, 2016

```
### | Original Line
```

The line number should be right aligned in three spaces, be followed by a space, then a pipe, another space, and lastly the original line. For example:

Input File:

```
This is a line of text
Yet another line
You only live twice
⋮
This is the last line
```

Output File:

```
  1 | This is a line of text
  2 | Yet another line
  3 | You only live twice
⋮
123 | This is the last line
```

You may assume that no file will have more than 999 lines.

Test your code against `LineNumbersTest.java`. When all tests pass proceed to the next exercise.

Exercise 1 Complete

Run:

```
git add .
git commit -m "Completed exercise 1"
git push origin master
```

2.2 Cookie Jar

Exercise 2

Create a class named `CookieJar` and implement the following method in it.

- `public static void cashingIn(File input, File output)`

  This method receives an input file with your count of coins given as pairs of integer and type of coin, e.g., `1 penny`, `42 quarters`, `9 nickels`. The method reads number/coin-type pairs from the input file and write the total dollar amount to the output file. For example, given the input file `2 quarters 4 dimes 1 penny 3 nickels 3 pennies`, the method prints `You have $1.09 in the jar` to the output file. The dollar amount should be formatted in 2 decimal places and with commas (when appropriate). In cases when the total amount in the jar is zero the method writes

  Code:

  ```
  You have no money in the jar
  ```

  to the output file. The input file may contain coin pairs in one line or across different lines, may contain several counts of the same type of coin at different times, and may list the type of coin in plural or singular form, e.g., `pennies` for many 1 cent coins or `penny` for one coin. Pennies are worth 1 cent each, nickels are worth 5 cents each, dimes are worth 10 cents each, and quarters are worth 25 cents each.

  You may assume that if you have an integer in the file, then a coin-type will follow it.

Test your code against `CookieJarTest.java`. When all tests pass, proceed to the next exercise.

Exercise 2 Complete

Run:

```
git add .
git commit -m "Completed exercise 2"
git push origin master
```

---

3 Common Mistakes

The following are warnings about and solutions to common mistakes for this lab.

1. Be sure to close your `Scanner`s and `PrintWriter`s!
2. When iterating through a `File`'s contents, ensure that you do not go past the end of the file.
3. Pay close attention to the format of your output.
4. Ensure that you handle any exceptions that your code may throw.