CS1003 Week 2 Exercise 2: Java I/O and Exceptions

As with all lab exercises, this exercise is not assessed. It is intended solely to help you understand the module material. There is probably more here than you will have time for during the lab hour; you are encouraged to complete it in your own time if you don't finish it.

The aim of this exercise is to practice Java file I/O and exception handling.

This exercise builds on the previous one, this time focusing on writing text to files.

1. Writing to a file

Make a copy of your solution from the previous exercise. Change it so that it writes out the contents of the input file to a separate output file. For this exercise, you could either hard-code the file names into your program or use multiple command-line arguments, one for input file name and one for output file name, this is up to you.

Test your program on the short and long test files. At this stage the contents of the input and output files should be identical. You can check this using the command `diff` *`file1 file2`*. It will give no output if it can't find any differences.

2. Handling exceptions

Now re-run your program, but get it to attempt to read a file that does not exist. You can do this either by specifying the name of a non-existent input file on the command-line (or in your code) or by moving the input file to some other folder. See what it does when the input file cannot be found in the expected location. Now make a copy of your previous program, and modify it so that it prints out a friendly error message if the input file cannot be found.

Try removing read permission from the input file with the command `chmod u-r test-short.txt`. Again, test your program's behavior and adapt if necessary.

3. Custom exceptions

Make a copy of your previous program, and modify it so that a new exception called *EmptyFileException* is thrown if the input file is empty.

4. Processing file data

Make a copy of your previous program, and modify it so that it removes all vowels from the text in the output file.

To get started, write two helper methods with the following signatures:

```
String removeVowels(String s) // which should return a copy of the string s with the vowels removed

boolean isVowel(String character) // which should return true if the character is a vowel.

// What should you do if the String is not a single character?
```

Now use these in your main program.

5. Testing

Modify your program so that it counts the characters in the output file and prints out the result at the end. Test your program on short and long files. The number of characters should be 191 and 67,370 respectively.

Check your character counts against the result given by the word counting command `wc` *`file`*.

6. Submission

You are required to submit a zip file containing your exercise attempt to the appropriate slot in the Exercises category on MMS.