**CP122 - Lab #8 - File I/O: Redaction**

**Assignment Overview**



Computers are really good at doing repetitive tasks quickly. This allows them to process a lot of information in a small amount of time. In today's assignment, our programs will process data in text files and write out new files. This is exciting because this is the first time that we can really start to see how useful our programs can be. The same small program can be used to process a file with 10 lines of data or a file with 10,000,000 lines of data.

**Cleaning Up Private Documents (Redaction.java)**

Governments, law firms, and businesses often have sensitive documents with private information. Occasionally these documents may need to be distributed more widely than was first intended. This often requires removing names of particular people to protect their privacy.

Your program for today needs to carefully remove all occurrences of a certain name from a given document. Your program should traverse the document and search for names to remove. Then the document should be re-written to a new file with each name occurrence replaced with the word "*REDACTED*".

Your program should prompt the user to enter three Strings:

* The filename of the original document
* The name to be removed and replaced
* The output filename where the cleaned document will be stored

Use the Scanner class to read in the original document and use the PrintWriter class to write out the cleaned version. Try testing your redaction program on a public domain novel from [Project Gutenberg](http://www.gutenberg.org/wiki/Main_Page). Download a .txt version of a novel you like and try replacing a character's name using your program.

Make sure your program handles possible exceptions cleanly.

**Discussion Questions:**

* What is the difference between catching an exception and throwing one? When should you use each construct?
* What are the possible exceptions that can occur when opening and reading from a file? How did you handle them in your program?
* Once you have your Redaction.java program working, test it on a couple of free books from Project Gutenberg. Test it on books of different lengths and record roughly how long it took your program to finish running. Can you think of a simple mathematical formula that describes the running time of your program as it relates to the length of the input book?