CS 570: Programming Foundations
Fall 2017

Homework Assignment #7

Due date: Wednesday, November 29, 2017 by 11:59 PM.

Note: This and all assignments given in this course can be and must be solved using **only** the materials that have been discussed in class. Do not look for alternative methods that have not been covered as part of this course.

Program (100 points):

In this assignment, we are going to practice reading from a file and writing the results of your program into a file. You will also practice catching exceptions. To that effect, we will "redact" specific words on a data file.

Program details

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.

You are to write a program that removes all occurrences of a certain name from an input text file. Your program should traverse the document and search for the name to remove. Then the document should be re-written to a new text file with each name occurrence replaced with the word "REDACTED".

Your program should prompt the user to enter three `String`s:

- The filename of the original document (this is your input file)
- The name to be removed and replaced (this is a string and it could be one or more words. For example John Smith)
- The output filename where the cleaned document will be stored (this is your output file)

Make sure your program handles possible exceptions cleanly.

Hint: You will need to review String methods discussed in week 3 this term. In particular, you may find the `indexOf` and `substring` methods very helpful:

indexOf

```
public int indexOf(String str,
                   int fromIndex)
```

Returns the index within this string of the first occurrence of the specified substring, starting at the specified index. The returned index is the smallest value *k* for which:

```
k >= fromIndex  && this.startsWith(str, k)
```

If no such value of *k* exists, then `-1` is returned.

Parameters:

`str` - the substring to search for.

`fromIndex` - the index from which to start the search.

Returns:

the index of the first occurrence of the specified substring, starting at the specified index, or `-1` if there is no such occurrence.

substring

```
public String substring(int beginIndex,
                        int endIndex)
```

Returns a string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex - 1`. Thus the length of the substring is `endIndex-beginIndex`.

Examples:

```
"hamburger".substring(4, 8) returns "urge"
"smiles".substring(1, 5) returns "mile"
```

Parameters:

`beginIndex` - the beginning index, inclusive.

`endIndex` - the ending index, exclusive.

Returns:

the specified substring.

Throws:

`IndexOutOfBoundsException` - if the `beginIndex` is negative, or `endIndex` is larger than the length of this `String` object, or `beginIndex` is larger than `endIndex`.

Try testing your redaction program on a public domain novel from Project Gutenberg. Download a **`.txt`** version of a short novel you like and try replacing a character's name using your program. Alternatively, you can also create your own input file by using a text editor (remember to save your file as a **`txt`** file). You can then go to your favorite news site, find an article, and copy and paste the text of the article into your **`txt`** file.

Here are a couple of sample runs of a typical solution to this problem:

```
Enter the name of the data file: mydata.txt
Enter the name of the output file: out.txt
Enter the name to be redacted: Superman


File was not found or could not be opened.

All done! Check your output file.
```

```
Enter the name of the data file: myFile.txt
Enter the name of the output file: redacted.txt
Enter the name to be redacted: Batman


All done! Check your output file.
```

Input File sample:

```
A year and a half ago, Batman v Superman: Dawn of Justice wasn't just a disappointing superhero movie. It became a meme, the new symbol of everything that could go wrong in a Hollywood comic-book spectacle. It was ponderous and inflated, its logic didn't parse, it had a Batman whose husky glower made him seem a stand-in for other (better) Batmans, it had a villain who was a jittery basket-case cliché — and more than that, it oozed the kind of solemnly overripe darkness" that was meant to signify integrity but was, in fact, laid on with a corporate trowel. Kiss of death: The movie underperformed at the box office.

It should be noted that the meme was overstated. Batman v Superman was written off as a commercial disappointment when, in fact, it did just fine. (Ditto forSuicide Squad, the lousy superhero movie that saved August.) And some of us believed that the crucifying ofBvS was a bit extreme; I liked the film's first half (before it went off the rails), and thought that the malevolent-Superman plot resulted in Henry Cavill giving his first wide-awake screen performance. The meme, however, became a myth, and you'd better believe that the studio gods paid attention.
```

Output File sample:

```
A year and a half ago, REDACTED v Superman: Dawn of Justice wasn't just a disappointing superhero movie. It became a meme, the new symbol of everything that could go wrong in a Hollywood comic-book spectacle. It was ponderous and inflated, its logic didn't parse, it had a REDACTED whose husky glower made him seem a stand-in for other (better) REDACTEDs, it had a villain who was a jittery basket-case cliché — and more than that, it oozed the kind of solemnly overripe darkness" that was meant to signify integrity but was, in fact, laid on with a corporate trowel. Kiss of death: The movie underperformed at the box office.

It should be noted that the meme was overstated. REDACTED v Superman was written off as a commercial disappointment when, in fact, it did just fine. (Ditto forSuicide Squad, the lousy superhero movie that saved August.) And some of us believed that the crucifying ofBvS was a bit extreme; I liked the film's first half (before it went off the rails), and thought that the malevolent-Superman plot resulted in Henry Cavill giving his first wide-awake screen performance. The meme, however, became a myth, and you'd better believe that the studio gods paid attention.
```

Note: Please make sure to submit well-written programs for these programming tasks. Good identifier names, useful comments, indentation, and spacing will be some of the criteria that will be used when grading this assignment.

Grading

Criteria for the Program	Points
Code Compiles Correctly – no compilation errors	20
Code runs properly on several tests and produces the correct results	20
Program Construction: all the required parts are there and logical flow is correct, including handling exceptions, and file I/O.	25
Correct implementation files for input and output	10
Correct implementation of strings	10
User Interface: the output of the program is clear and well organized, easy to read and understand.	5
Code Style: good identifier names, named constants as needed, comments, indentation and spacing	10

How to submit your assignment

- Assignments must be submitted via Blackboard Learn.
  - Please note that assignments submitted via email will not be accepted.
  - Late assignments will not be accepted. Your work must be uploaded and submitted by 11:59 PM on the date it's due.

- For this assignment you must submit:
  - A `.java` file for your source code.
  - Do not submit files in any other formats – if you do, your assignment will not be graded.

Academic Honesty

You must be the sole original author of the solution you submit. You must compose all program and written material yourself. All material taken from outside sources must be appropriately cited.