Introduction to Computer Science 2

Lab 6: Exception Handling and Streams

Learning Goals:

- To learn how to use exception handling.
- To learn how to process streams and files.

Exercise 1 (3 points)

Write a class `FileCounter` (template provided) that asks a user for a file name and keeps track of the number of characters, words, and lines in that file.

First users of your class call the following method on a single file:

```
public static void processFile(String fileName)
```

After which, the following methods should return the word, character and line counts of the given file:

```
public static int getWordCount()

public static int getCharacterCount()

public static int getLineCount()
```

You can use the provided `FileCounter.java` as a starting point and the files: `file1.txt`, `file2.txt` and `file3.txt` to test your program.

Exercise 2 (3 points)

Write a class Find.java (template provided) that searches all files specified on the command line and prints out all lines containing a reserved word given as the first parameter.

First write the method:

```
public static String[] getLinesWithWordForFile(String file, String word)
```

Which returns an array of lines, each containing the String word.

Next, write a main method that uses command line arguments to handle multiple files and a single **word** to find the given word in all lines in all of the assigned files.

If you want to test your class, you need to pass some parameters in the command line. To that end, you must compile it in advance. You can use, for example:

```
javac Find.java
java Find ring report.txt address.txt Homework.java
```

Then the program should print:

```
report.txt: has broken up an international ring of DVD bootleggers
address.txt: Kris Kringle, North Pole
address.txt: Home Simpson, Springfield
Homework.java: String filename;
```

Note: If you use ArrayLists to store the partial solutions, you may find the following link helpful to complete this exercise.

Convert ArrayList to Array

Exercise 3 (4 points)

You should modify the `BankAccount` class (provided as a template) to throw an `IllegalArgumentException` when the account is constructed with a negative balance, when a negative amount is deposited, or when the amount that is not between 0 and the current balance is withdrawn.

Write a test program that causes all the three exceptions to occur and that catches them all.

Honor code, coding style, and deliverable:

Try to solve the exercises with what you already know. You are welcome to expand your program to do extra things but they are not mandatory.

Plagiarism is not allowed! We will run sophisticated software that automatically detects similarities on source code among students. All plagiarism incidents will be immediately reported to the Board of Examiners

Submission!

Submit your java files to CodeGrade (see more information in the Syllabus).
Ask your instructor in case there is a problem with your submission.

DO NOT SEND SUBMISSIONS VIA EMAIL
YOUR LAB WILL NOT GET GRADED!