Programming Challenge for S&CA Candidates

Below are instructions for the programming challenge. Please confirm that you have received this email, and let us know your estimated delivery time. If you have any questions, let me know. Once you complete this challenge, please send it back with some instructions on how to run it. Thanks for your time!

We would like you to write a class using Groovy to search and replace within all files present in a given directory and subdirectories underneath. Include comments in your class to indicate why you chose the specific algorithms you did (especially anything you found particularly cool).

If you have never used Groovy before, that is ok; there is a lot of information about the required techniques available on the internet and you will not be penalized for borrowing the technique from others. If you are very familiar with Groovy, then we expect to learn something from your exercise.

There is a great "getting started" article that includes setting up the complete environment here:

http://groovy-lang.org/learn.html

We expect you to spend some time thinking about, and completing, the exercise. Please record how much time it takes you to complete.

1. The Program will require AT A MINIMUM these arguments:
    1. A path to a directory containing some text file (you can assume that the directory contains only text files)
    2. The original text or pattern to use for searching in the files (in the given directory or its subdirectories).
    3. A new text string which will replace the original text or pattern if found in the file (as many times as it was found).
2. The class will also allow another optional argument for a path to a file for outputting a list of which files were modified (logging).
3. Add comments to your class.
4. Extra points if you back up the original file before replacing the text.
5. Extra points if you implement some simple and creative logging (start time, errors, end time, pattern found and where, etc.)
6. **IMPORTANT NOTE: the program must receive all parameters through command line arguments (no argument hardcoding), and make no assumptions about environment conditions (i.e. avoid "runs in my machine")**