Java Coding Exercise - 1 hour

Instructions for the candidate

The exercise is to write a command line driven text search engine, usage being:

```
      java mainClassFile pathToDirectoryContainingTextFiles
```

This should read all the text files in the given directory, building an in memory representation of the files and their contents, and then give a command prompt at which interactive searches can be performed.

An example session might look like:

```
$ java -jar SimpleSearch.jar /foo/bar
14 files read in directory /foo/bar
search>
search> to be or not to be
filename1 : 100%
filename2 : 95%
search>
search> cats
no matches found
search> :quit
$
```

I.e. the search should take the words given on the command prompt and return a list of the top 10 (max) matching filenames in rank order, giving the rank score against each match.

Note: treat the above as an outline spec; you don't need to exactly reproduce the above output. Don't spend too much time on input handling, just assume sane input.

Ranking

- The rank score must be 100% if a file contains all the words
- It must be 0% if it contains none of the words
- It should be between 0 and 100 if it contains only some of the words - but the exact ranking formula is up to you to choose and implement

Things to consider in your implementation

- What constitutes a word
- What constitutes two words being equal (and matching)
- Data structure design: the in memory representation to search against
- Ranking score design: start with something basic then iterate as time allows
- Testability

Deliverables

- Code to implement a version of the above
- A README containing instructions so that we know how to build and run your code

Example starting point:

```java
import java.io.File;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        if (args.length == 0) {
            throw new IllegalArgumentException("No directory given to index.");
        }
        final File indexableDirectory = new File(args[0]);
        //TODO: Index all files in indexableDirectory
        Scanner keyboard = new Scanner(System.in);

        while (true) {
            System.out.print("search> ");
            final String line = keyboard.nextLine();
            //TODO: Search indexed files for words in line
        }
    }
}
```