COSC 310 – Project 2

Spelling Checker

Goal: Implement the Spell Check class as described on page 457 of Data Structures & Algorithms in Java (P-10.74).

Description: Create a class that can attempt to return the correct spelling of given words. You will use a Hash-based Set. A Set is simply a map in which keys do not have associated values. (See text, Chapter 10). Your program should check a number of common misspellings as detailed below, as well as common phonetic substitutions. You will need a spell-check module, associated data structures, and a driver class that tests your program.

Deliverables:

1. Program source code – Use edu.frostburg.cosc310 as the base package for your code.
2. Runnable .jar file that shows runs your test. It should clearly demonstrate that your spelling checker is functioning (but you do not require a graphical visualizer).
3. Code style – Your code will be professionally written.
4. Documentation – Describe all the important parts of your code. You should not assume I know anything about your assignment.
5. Readme.txt – A short file that describes how to start your program (and not much else).
6. Post-mortem – PDF file describing the progress you made on your assignment. In particular, detail what worked and what did not. The goal is to organize what you learned from your code.

Tips:

- *This is not a short assignment*. Do not start it at the last minute.
- Good design is good design.
- Plan out your program. Use pseudo-code, UML, every tool that you have available.
- You need your textbook.
- Feel free to ask me for help, but I will only help you if I see that you are making progress already.

Checklist:

Check	Component	Grade %
	Spelling Checker Module - Set - Swapping letters - Inserting wrong letters - Missing characters - Wrong characters - Phonetic substitution	65%
	Driver class	10%
	Code Style (clean and professional)	10%
	Documentation	5%
	Post Mortem	5%
	Deliverables in a zip: 1. Source code 2. YourProgram.jar 3. Readme.txt 4. Post-Mortem	5%

Avoid list:

Check	Component	Grade %
	Compilation Errors	-50+%
	Typos	-10%
	Bugs	-x%