Rubric for Assignment #3

Criteria		Unsatisfactory	Acceptable	Good	Exceptional	Marks
		0	1	2	3	
Functionality	Binary Search Tree	- BST isn't used to store dictionary - no attempt to balance tree was made - tree was not written to a file - too many errors exist	- dictionary was correctly read into BST - BST was written to a file - an attempt to balance tree was made and can be explained - some errors exist	- dictionary BST was read and the output written to a file correctly - BST has almost all nodes balanced - a few errors exist	- BST contains all dictionary words and is perfectly balanced - balance algorithm can be explained - can easily view tree in a file	x5
	Spell Check Output	- the program does not output misspelled words from the test document - too many errors in output exist	- some misspelled words are correctly identified and listed - could not handle punctuation, capitalization or special characters - some errors exist	- most misspelled words are correctly identified and listed - could not handle rare cases - a few errors exist	- all misspelled words in the text document are correctly identified and listed - no errors exist	x5
					Sub-Total	30

Note : 30 x 25% = 7

Note: The following aspects of the program will only be graded if you receive more than 25% on the functionality rubric.

Output	**Aesthetics**	- incorrect or not existent use of whitespace in output - output is confusing and hard to follow	- fair use of whitespace - most output is clear, but poorly presented	- good use of whitespace - output is clear and fairly well presented	- excellent use of whitespace - output is clear and attractively presented	_________

Source Code	**Readability**	- source code is poorly organized and very difficult to read	- source code can be read, but is hard to follow	- source code is fairly easy to read, but is hard to follow in some areas	- source code is exceptionally well organized and easy to follow	_________
	Reusability	- source code cannot be reused - no functions or classes used	- small sections of code could be reused	- large portions of code could be reused with some modifications	- source code could be easily reused with little modifications	_________
	Efficiency	- contains large portions that could have been easily reduced using a different method - a lot of code is duplicated, copy/pasted	- tried some methods to improve efficiency - can explain what they attempted	- employed good ideas to improve efficiency - can point out where other improvements could be made	- very clean and efficient code - can propose new ideas for improvement	_________
	Comments	- little to no comments used	- comments are used, some are meaningful and easily understood - some files and functions have headers	- comments are used extensively, most are meaningful and easily understood - most files/functions have headers	- not over/under commented - comments are meaningful and easily understood - files/functions have headers - is self-documenting	_________
	Naming Convention	- no standard naming convention followed	- a standard naming convention was used for part of the program, but deviated often	- a standard naming convention was used for most of the program and deviated very little	- industry standard naming convention used throughout the program	_________
	Consistency	- no consistency in formatting or layout of source code	- source code formatting was fairly consistent, but contained some inconsistency with whitespace, brackets, etc	- source code formatting was very consistent with respect to whitespace, brace brackets, parentheses, etc	- source code formatting never deviated from the programmer's layout	_________

Sub-Total	_________ **21**
Total	_________ **51**

Assignment: ___________________________