**Line-Based File I/O: Finding A Word**

**The goal for this exercise** is to make sure that you can open up a file of text textual tokens (i.e., words), search through it, and then find a particular word.

For this exercise you will process a file that contains a single word on each line, such as:

Laptop

Laser

Macho

Sam

Samantha

Mulder

Microphone

Aardvark

Notice that the words in the file are not arranged in any particular order. You may assume that each word is the ONLY thing on each line, and that there is no extra spaces either before or after the word.

Your program is to give an introduction and then prompt the user for a word to search for. Then it will read through the data file searching for that word. If it finds it, it print out a message telling the user that the word was found; if the word is not present in the file then the program should print a different message telling the user that the word is absent.

 Your program should consider the user’s input to be a token, and match it against the text tokens (words) in the file. In other words, tour program should only report a match if the user’s input completely matches the word. For example, if the user types in “Sam”, your program should match “Sam”, but NOT match “Samantha”

 Your program must do this in a case-insensitive manner (so “Sam”, “sam”, “sAM”, etc, would all match the “Sam” listed in the above file)

Hint: Remember that you can get each line of the file in turn by calling ReadLine.

**What you need to do for this exercise:**

1. Implement the method FindWord (in the class named File\_Exercises) so that it behaves as described above
	1. Your function must be able to read the files located in the “Files\Exercise\_Files” subfolder. While you’re obviously free to rearrange the files on your personal computer while you’re developing the various parts of this exercise you are required to be able to read the file named “Files\Exercise\_Files \Words.txt” when you hand in the exercise
	2. You should compare the names in a case-insensitive manner.
	Hint: Look back to prior lessons for a method on the String class that will do a case-insensitive comparison of two strings for you.