CSE 382
Exercise - C# IO and Data Structures

You will submit your work via Git. You will embed comments into your code:

- For each method you write, put a comment immediately above the method indicating whether it is complete or not. This can be as simple as: // This method is complete and correct.
- **Answers to the bold questions are to be put at the top of IO.cs**

Make sure that you commit and push your changes to the CS project. There will be nothing submitted to Canvas.

Make the following changes in IO.cs

- Create the following methods, which all do essentially the same thing. Each method opens the text file of strings (one per line) and creates a collection to store them.
- public static List<string> ReadWordsList(string fileName) {
  List<string> result = new List<string>();
  // Fill in
  return result;
  }
- public static SortedSet<string> ReadWordsSortedSet(string fileName) {
  SortedSet<string> result = new SortedSet<string>();
  // Fill in
  return result;
  }
- public static HashSet<string> ReadWordsHashSet(string fileName) {
  HashSet<string> result = new HashSet<string>();
  // Fill in
  return result;
  }
- Test your methods with the file "words.txt"
  - **Was there any substantive difference in computing time between them?**
    - The System.Diagnostics.Stopwatch class can be useful for timing pieces of code.
    - Use stopwatch.Restart() when timing a second part of code.
    - If the times produced are really small (e.g., 0.0001), call the methods several times so that the time is, say, at least 3 seconds long.
- Looking up from dictionary
  - Write a loop to determine count the number of legal words in the following array:
    string[] words = { "COMPUTERS", "ZYMOTIC", "AARDVARK", "WORD", "DATABASIC" };
  - Use the Contains method for all three data structures: List, SortedSet, HashSet.
  - **Was there a substantive time difference in any of the data structures?**
    - It might help to repeat the searches, say, 100, times to ensure that there is an appreciable amount of time spent.
- Iterating over a collection
  - Use the following foreach as the basis for determining how many legal words contain exactly 3 letters. **How many 3 letter words are there?**

- `List<string> L = ReadWordsList("words.txt");`
- `foreach (string word in L) { … }`