CSC240 – SPRING 2013
ASSIGNMENT #1

Review of O-O Concepts
Text Processing

Congratulations! You have been hired by Wholly Nutritious Foods (however, it would be an ethical violation to put this in your resume'). Wholly Nutritious Foods is a nation-wide food chain that sells wholly nutritious foods (as you might have guessed). Your four assignments during the semester will relate to data that relates to the wonderful wholly nutritious food products that we sell.

Much of the material for this assignment is review material from CSC142 with some new text processing stuff (needed for the nutrients field in the FrozenFood class). (You can either use the String split method or the StringTokenizer class to convert the nutrients field described below to a collection of tokens that can be displayed separately.)

This first assignment involves keeping track of frozen foods that our company (WNF) sells. The FrozenFood class has the following fields:

Field	Type	Description
name	String	Name of the frozen food product (e.g., "VEGGIE BURGER")
manufacturer	String	Name of the company that makes this frozen food product ("FRESH GARDEN")
nutrients	String	A list of the top nutrients that this frozen food provides (e.g., "PROTEIN, SODIUM; VITAMIN A")

Your program will run in two phases:

- data entry – when you enter the data for the frozen food products
- sorting and searching – when you sort the data alphabetically by the names of the frozen food products and then search for data relating to those products using various search criteria.

Data Entry

During the data entry phase of your program, the user will be asked to enter the data for the frozen food products in a loop. The data for at least one frozen food product will be read in. After the data for a frozen food product is read in, the user is asked if he/she wants to enter data for another frozen food product. If the user responds N (for NO), then the data entry phase terminates.

The data that the user enters will be stored in an array of references to FrozenFood objects. A good strategy would be to create a relatively large array to store the references in as the program goes through the loop described in the previous paragraph. Keep a count of how many FrozenFood objects were created and read in. When the user is done entering the data, you can then use the count (like numFF) to create a new array. You can copy the data in the original array into the new array. This new array will be "filled to the gills" (as I like to say). Thus, you won't need to pass the number of FrozenFood objects as a parameter to the various methods you will be creating. Instead, the methods can apply .length to the parameter array (in these methods) to determine the number of objects in the array.

For each FrozenFood product, the user will be asked to enter the name of the FrozenFood and its manufacturer. Then, the program will ask the user for the number of nutrients that the user will be entering for this product. Then, the user will be asked to enter the names of those nutrients. For example, in the VEGGIE BURGER example given in the above table, the nutrient data might be entered as follows. User inputs are shown in boldface and underlined:

NUMBER OF TOP NUTRIENTS FOR THIS PRODUCT? **3**

ENTER THE TOP NUTRIENTS WHEN PROMPTED.
#1: **PROTEIN**
#2: **SODIUM**

```
 #3: VITAMIN A
```

Thus, the entire data entry process for this FrozenFood product might look like this:

```
NAME:  VEGGIE BURDER
MANUFACTURER:  FRESH GARDEN

NUMBER OF TOP NUTRIENTS FOR THIS PRODUCT?  3

ENTER EACH OF THE TOP NUTRIENTS WHEN PROMPTED.
 #1: PROTEIN
 #2: SODIUM
 #3: VITAMIN A
```

Sorting and Searching Phase

Immediately after the completion of the data entry phase, your program will sort the FrozenFood data alphabetically by the names of the frozen food products using bubblesort. Then, you will present the user with the following searching menu:

1 – Display the names of all frozen food products
2 – Display all data for a specific frozen food product
3 – Display the names of all frozen food products from a given manufacturer
4 – Display the names of all products with a given top nutrient
5 – Quit the program

Your program should be robust enough to allow the user to enter an illegal menu choice (such as '9' or 'z'). If an illegal value is entered for the menu choice, the user must be informed of this. Then, the user will be presented with the menu once again.

If the user enters '5' to quit, the program must give the user the opportunity to confirm that he or she actually wants to quit:

```
    Are you sure (Y/N)?
```

If the user enters Y or y, the program terminates. Otherwise, the user is returned to the searching menu.

If the user chooses any option '1' through '4', that option is processed (as described below) and after that option is processed, the program will again display the searching menu in order to allow the user to enter another choice. What follows are descriptions of the menu choices '1' through '4'. (Menu choice '5' was described above.)

If the user chooses '1' from the menu, the names of all the WNF frozen food products will be displayed in alphabetical order. Note that the names must be capitalized so that the alphabetical ordering of the data will not be case-sensitive.

If the user chooses '2' from the menu, the program will ask the user to enter the name of a specific frozen food product. The program will then either display the name of the manufacturer and the list of top nutrients for that frozen food product or it will provide feedback to the user to the effect that this FrozenFood object was not found in the list. **The program must use the binary search algorithm to perform the search.** (A pre-condition for the use of binary search is that the data is sorted alphabetically by name). The search must not be case-sensitive. Thus, "VEGGIE BURGER" and "Veggie Burger" are one and the same insofar as the program is concerned.

Suppose the user asks to see the data for the Veggie Burger product (by entering "VEGGIE BURGER" when prompted for the name of the product that the program will search for). The Veggie Burger data might be displayed as follows:

```
    MANUFACTURER:  FRESH GARDENS
    TOP NUTRIENTS FOR THIS PRODUCT:
      PROTEIN
      SODIUM
      VITAMIN A
```

If the searched for FrozenFood object does not appear in the array, the user is given appropriate feedback.

If the user chooses '3' from the menu, the program will ask the user to enter the name of a manufacturer. The program will then search through the array and list the names of all of the frozen food products that were produced by that manufacturer. If no FrozenFood objects have that manufacturer listed as its manufacturer, the user will get appropriate feedback.

If the user chooses '4' from the menu, the program will ask the user to enter the name of the nutrient that the user is interested in. The program will then go through the array of FrozenFood objects and search through the nutrients field for each FrozenFood object to see if the specified nutrient is one of the top nutrients listed for that frozen food product. For each FrozenFood object that does have that nutrient listed in its nutrients field, the name of that FrozenFood will be displayed. If no FrozenFood object has that nutrient listed in its nutrients field, the user will be given appropriate feedback.

grading

Your program will be graded using the grading sheet that will be posted on D2L. Your program should satisfy the items in the grading checklist as much as possible. (Hopefully, 100%.) In addition, your program will be graded on:

- quality of the documentation
- clarity of your code
- the use of appropriate conventions of style
- correct use of classes and objects
- timely submission of your work
- submission of a project plan document