Due before: **11:59 PM on Wed Apr 20 & Apr 27 2022**

CSE-271: Object-Oriented Programming

Homework #5: Cookbook

Phase #1: Wed Apr 20 2022 before 11:59 PM

Phase #2: Wed Apr 27 2022 before 11:59 PM

Email-based help Cutoff: 5:00 PM on Tue before deadline

Delayed (by no more than 24-hours) submissions earn only 80% credit

Maximum Points: 24

Key objectives of this project are:

- Gain experience with processing text files
- Gain experience with working with binary files
- Review concepts of creating custom classes
- Review and adhere to CSE department's Style guide
- Use Javadoc to document methods and their return values

Submission Instructions

Each phase of this project must be turned-in electronically via Canvas CODE plug-in. Ensure your program compiles successfully, without any warnings or style errors. Ensure you have documented the methods. Ensure you have tested operations of your classes.

General Note: Upload each file associated with homework (or lab exercises) individually to Canvas. Do not upload archive file formats such as zip/tar/gz/7zip/rar etc.

Grading Rubric:

The source code submitted for this homework **must pass necessary test(s) in order to qualify for earning any score at all**. Programs that do not meet basic requirements or just skeleton code will be assigned zero score! Programs that do not compile, **have even 1 method longer than 25 lines**, or just some skeleton code will be assigned zero score.

- **NOTE:** Violating CSE programming style guidelines is an error! Your program should not have any style errors.
- **Delayed submission: Only 80% points:** Submission delayed by no more than 24-hours will be accepted for a partial credit of maximum 80% of the points.
- **Conciseness, Formatting & Documentation: 2 points** – Reserved for concise solution, good style and formatting in each phase. These points may be the hardest to earn!
- **Phase #1**: Implement `Ingredient.java` and 2 methods in `Recipe.java`: **10 points**
- **Phase #2**: Implement methods in all the classes: **10 points**

Project Overview:

In this assignment you will be using an `Ingredient` class to build a `CookBook` class consisting of a list of `Recipe` objects. A `Recipe` consists of a one or more `Ingredient` objects. A class to call various methods in `CookBook` to help test its functionality is supplied with this project.

Starter files provided

In order to help you get jump started on the project the following files are provided. Note that files in Red are the files that you need to modify and turn-in as a part of this project. Rest of the files should not be modified and need not be turned-in.

File Name	***Description***
`Ingredient.java`	A simple class that represents an ingredient in a recipe. A recipe consists of one or more ingredients. The starter code containing the methods that you need to implement.
`Recipe.java`	This class essentially encapsulates a set of `Ingredient` objects maintained as an array. The starter code contains the methods you need to implement.
`CookBook.java`	This class represents a cook book that consists of an array of `Recipe` objects.
`InvalidIngredientException.java`	A simple, custom exception class to be used in this exercise to report an application specific error.

`badrecipe.txt`
`biscuits.txt`
`cranoatmeal.txt`
`jerusalemsalad.txt`
	Some sample input text files that contain ingredients for a recipe. These files are used for testing your program.

`CookBookMenu.java`	The main testing program that provides an interactive mechanism to test the functionality of various methods in `CookBook.java` and `Recipe.java` classes.

The supplied skeleton classes for `Ingredient.java`, `Recipe.java` and `CookBook.java` already contain all the necessary instance variables. Consequently, you **are not permitted** to add any additional instance variables. If you add any instance variables to these two classes you will be assigned zero grade.

Review format of supplied recipe files

View one of the recipe files (say: `biscuits.txt`) in `Eclipse` by simply double-clicking on the file. Observe that the first line of the file contains a title for the recipe, and the remaining lines contain the ingredients used by the recipe (each line contains a single ingredient). The ingredients are divided into three components: *quantity*, *measurement*, and description. To know which part of the line is which component, the format of the line is as follows: *Quantity***;** *Measurement***;** *description*.

If *quantity* is empty assume that it is 1.0, and if measurement is empty assume that it is "" (empty string). Example ingredients are as follows:

Line in recipe text file	*Interpretation*
`.5; cup; All Purpose Flour`	`0.5 cup of All Purpose Flour`
`;;beet`	1 beet
`;gallon; Milk`	1 gallon Milk

Refer to the provided recipes (`biscuts.txt`, `cranoatmeal.txt`, and `jerusalemsalad.txt`) for additional examples.

Phase #1: Develop `Ingredient` and part of `Recipe` class

In the first phase of this project, you will be developing the following two classes:

1. `Ingredient.java`: Develop all of the methods
2. `Recipe.java`: Develop just the `addIngredient` and `createRecipe` methods (rest of the methods will be developed in Phase #2).

Phase #1 Tasks #1: Create a new Java project & download starter code

1. Create a Java project in `Eclipse` and download all of the starter files directly into your project and refresh your project.
2. Review the starter code for the different classes to form a good mental model of the project prior to starting to work on it.
3. Review the source code in `CookBookMenu`.java class to understand how it uses `CookBook.java` and `Recipe.java` classes that you will be developing.

Phase #1 Task #2: Implement methods in `Ingredient.java` and `Recipe.java`

There are a few methods that you need to implement. Most of the methods are relatively straightforward and reuse concepts from earlier exercises and assignments. So, use examples from earlier exercises to suitably implement the methods. In addition, you will need to tap into your basic logic skills for performing some routine array operations.

Rather than repeating the instructions for each method in this document, the necessary instructions regarding the functionality of each method have placed as Javadoc comments before each method. Read the Javadoc for the various methods in `Recipe.java` and suitably implement the methods.

Phase #1 Task #3: Testing

Once you have successfully implemented the methods you can test them by running `Phase1Tester.main()` and supplying inputs as shown in the sample outputs below. User inputs are in green.

Phase #1 Test Output #1:

```
Testing Ingredient.parseString method...
Done.
Enter recipe file name: biscuits.txt
Loading recipe from biscuits.txt
Recipe loaded:
Buttermilk Biscuits (from Foodnetwork.com)
2.0 cups flour
4.0 teaspoons baking powder
0.25 teaspoon baking soda
0.75 teaspoon salt
2.0 tablespoons butter
2.0 tablespoons shortening
1.0 cup buttermilk
```

Phase #1 Test Output #2:

```
Testing Ingredient.parseString method...
Done.
Enter recipe file name: cranoatmeal.txt
Loading recipe from cranoatmeal.txt
Recipe loaded:
Cranberry Oatmeal Chews
8.0 tablespoon butter
2.0 tablespoon oil
1.0 cup light brown sugar
1.0  zest of one orange
6.0 tablespoon sour cream
2.0 teaspoon vanilla
1.5 cup flour
0.5 teaspoon baking soda
1.0 teaspoon cinammon
0.5 teaspoon salt
2.0 cup oats
1.5 cup dried cranberries
0.5 cup walnuts
```

Phase #1: Submission via Canvas

This homework assignment must be turned-in electronically via Canvas CODE plug-in. Ensure your program compiles successfully, without any warnings or style errors. Ensure you have documented the methods with Javadoc. Upload the following to Canvas via the CODE plug-in:

- The 2 Java source files you have developed, namely: `Ingredient.java`, `Recipie.java`

Phase #2: Implement all methods

In this phase, you will be completing the cook book application by implementing all of the methods in `Recipe.java` and `CookBook.java`. Most of the methods are relatively straightforward and reuse concepts from earlier exercises and assignments.

Rather than repeating the instructions for each method in this document, the necessary instructions regarding the functionality of each method have placed as Javadoc comments before each method. Read the Javadoc for the various methods in `CookBook.java` and suitably implement the methods. **In addition, study the sample outputs below for more details**.

Phase #2 Testing

Once you have successfully implemented the methods you can test them by running `CookBookMenu.main()` and supplying inputs as shown in the sample outputs below. User inputs are in green. Outputs you need to pay attention to are highlighted.

Test Output #1 (**`search`** command):

```
Enter command (add,search,check,load,save,quit): search
Enter search string: teaspoon
No matching recipes found.
Enter command (add,search,check,load,save,quit): quit
```

Test Output #2 (**`check`** command):

```
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: blah.txt
Unable to load recipe because:
blah.txt (No such file or directory)
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: biscuits.txt
Searching for recipe Buttermilk Biscuits (from Foodnetwork.com)
Recipe not found.
Enter command (add,search,check,load,save,quit): quit
```

Test Output #3 (**`load`** command):

```
Enter command (add,search,check,load,save,quit): load
Enter cook book file name: badbook.dat
Unable to load cookbook because:
badbook.dat (No such file or directory)
Enter command (add,search,check,load,save,quit): quit
```

Test Output #4 (**`add`** command):

```
Enter command (add,search,check,load,save,quit): add
Enter recipe text file name: badrecipe.txt
Unable to load recipe because:
Each ingredient line must be of the form:
quantity ; measurement; description
Enter command (add,search,check,load,save,quit): quit
```

Test Output #5 (**`add`** and **`check`** command):

```
Enter command (add,search,check,load,save,quit): add
Enter recipe text file name: biscuits.txt
Recipe Buttermilk Biscuits (from Foodnetwork.com) added.
Enter command (add,search,check,load,save,quit): add
Enter recipe text file name: cranoatmeal.txt
Recipe Cranberry Oatmeal Chews added.
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: biscuits.txt
Searching for recipe Buttermilk Biscuits (from Foodnetwork.com)
Recipe was found in cook book
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: cranoatmeal.txt
Searching for recipe Cranberry Oatmeal Chews
Recipe was found in cook book
Enter command (add,search,check,load,save,quit): search
```

```
Enter search string: teaspoon
Buttermilk Biscuits (from Foodnetwork.com)
2.0 cups flour
4.0 teaspoons baking powder
0.25 teaspoon baking soda
0.75 teaspoon salt
2.0 tablespoons butter
2.0 tablespoons shortening
1.0 cup buttermilk

Cranberry Oatmeal Chews
8.0 tablespoon butter
2.0 tablespoon oil
1.0 cup light brown sugar
1.0  zest of one orange
6.0 tablespoon sour cream
2.0 teaspoon vanilla
1.5 cup flour
0.5 teaspoon baking soda
1.0 teaspoon cinammon
0.5 teaspoon salt
2.0 cup oats
1.5 cup dried cranberries
0.5 cup walnuts

Enter command (add,search,check,load,save,quit): quit
```

Test Output #6 (**`add`** and **`save`** commands):

```
Enter command (add,search,check,load,save,quit): add
Enter recipe text file name: biscuits.txt
Recipe Buttermilk Biscuits (from Foodnetwork.com) added.
Enter command (add,search,check,load,save,quit): add
Enter recipe text file name: cranoatmeal.txt
Recipe Cranberry Oatmeal Chews added.
Enter command (add,search,check,load,save,quit): add
Enter recipe text file name: jerusalemsalad.txt
Recipe Jerusalem Salad (from online) added.
Enter command (add,search,check,load,save,quit): save
Enter cook book file name: cookbook2.dat
Enter command (add,search,check,load,save,quit): quit
```

Test Output #7 (**`load`** and **`check`** commands):

```
Enter command (add,search,check,load,save,quit): load
Enter cook book file name: cookbook2.dat
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: biscuits.txt
Searching for recipe Buttermilk Biscuits (from Foodnetwork.com)
Recipe was found in cook book
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: jerusalemsalad.txt
Searching for recipe Jerusalem Salad (from online)
Recipe was found in cook book
Enter command (add,search,check,load,save,quit): check
Enter recipe text file to search: cranoatmeal.txt
Searching for recipe Cranberry Oatmeal Chews
Recipe was found in cook book
Enter command (add,search,check,load,save,quit): quit
```

Phase #2: Submission via Canvas

This homework assignment must be turned-in electronically via Canvas CODE plug-in. Ensure your program compiles successfully, without any warnings or style errors. Ensure you have documented the methods with Javadoc. Upload the following to Canvas via the CODE plug-in:

- The three Java source files you have developed, namely: `Ingredient.java`, `Recipie.java`, and `CookBook.java`.