CSE 114: Computer Science I

Homework #5

Spring 2017

Assignment Due: Monday, April 10, 2017 by 11:59 pm

Directions:

- Solve the following problems to the best of your ability.
- At the top of every file you write for this assignment, include the following information in a comment, with each item on a separate line:
  - your first and last name as they appear in Blackboard
  - your Stony Brook ID #
  - the course number (CSE 114)
  - the assignment name and number (Homework #5)

⚠ Your files, Java classes, Java methods, etc. must be named and/or defined as proscribed below. Work that does not meet the program specifications (e.g., wrong file names or wrong method names) will not be graded.

⚠ Upload your `.java` files to Blackboard by the indicated due date and time. Late work will not be accepted for grading. Work is late if it is submitted after the due date and time.

⚠ Source code that does not compile will not be graded.

⚠ Do not upload `.class` files. Such files be deleted.

⚠ Do not combine your `.java` files into a zip file, rar file or other archive. Such files will be deleted. The grader will not unpack archive files to search for your source code.

⚠ Do not include any `package` declarations in your source code unless directed to do so. Points may be deducted if your code must be edited to remove unnecessary package declarations.

Assignment Objectives

By the end of this assignment you should be able to design, code, run and test complex object-oriented programming and inheritance.

Part I: Recipe Book (20 points)

Filename(s): `Ingredient.java, RecipeIngredient.java, CookingRecipe.java, RecipeBook.java, TestRecipeBook.java`

In this assignment you will be creating a cooking recipes book. A RecipeBook will contains multiple CookingRecipe(s) and each recipe has multiple RecipeIngredient(s). To do this you will need to implement the following classes and methods:

Ingredient.java class:
has a constructor:

```
public Ingredient(String name, String measuringUnit, int caloriesPerUnit)
```

RecipeIngredient.java class is a subclass of Ingredient
has a constructor:

```
public RecipeIngredient(String name, String measuringUnit, int caloriesPerUnit,
float quantity)
```

CookingRecipe.java class
has the following constructors and methods:

```
public CookingRecipe(String name)
```

```
public void addOrUpdateRecipeIngredient(Ingredient ingredient, float quantity)
```

if the recipe already includes the ingredient specified by the parameter, then just update the quantity, otherwise add a new recipe ingredient.

```
public RecipeIngredient getRecipeIngredient(Ingredient ingredient)
```

```
public RecipeIngredient getRecipeIngredient(String ingredientName)
```

return the RecipeIngredient object corresponding to the ingredient object paramater, or return null if the ingredient is not part of the recipe.

```
public RecipeIngredient removeRecipeIngredient(Ingredient ingredient)
```

```
public RecipeIngredient removeRecipeIngredient(String ingredientName)
```

remove the given ingredient from the recipe. If the ingredient is part of the recipe return it, else return null.

```
public float calculateCalories()
```

calculates the sum of the calories for all the ingredients in the recipes and their respective quantities.

```
public int getNumberOfIngredients()
```

returns the number of ingredients in the recipe.

```
public String toString()
```

returns the recipe name and ingredients as a String.

The specification of this class intentionally does not specify the internal structure of the class (name, ingredients), so you can design it in any way you want, as long as you provide the functionalities required above.

RecipeBook.java class
should provide the following methods:

```
public RecipeBook(String bookName)
```

```
public CookingRecipe addRecipe(String name, RecipeIngredient[] ingredients)
```

adds to the book and returns a new cooking recipe with the given parameters. If recipe book already contains a recipe with the same name, do not create a new one and return null.

```
public CookingRecipe removeRecipe(String name)
```

removes the cooking recipe from the coocking book and returns it. If recipe book does not contain a recipe with the specified name, then return null.

```
public CookingRecipe[] findRecipes(RecipeIngredient[] ingredients)
```

returns all cooking recipes from the cooking book that contain all the ingredients passed as parameters. If recipe book does not contain any recipe with the specified ingredients, then return null.

```
public CookingRecipe[] findRecipesWithFewIngredients(int numberOfIngredients)
```

returns all cooking recipes from the cooking book that contain less then the number of ingredients passed as parameter. If recipe book does not contain any recipe with the specified number of ingredients, then return null.

```
public CookingRecipe[] findRecipesLowCalories()
```

returns all cooking recipes from the cooking book that have the lowest number of calories. Note: this can be multiple recipies.

All your above classes should implement the `toString()` and `equals(Object)` methods. The `toString()` method should return a string with the class name and then all the data fields one per each line. The Ingredient.java `toString()` method should return ``Ingredient\n''+ ``name='' + name + ``\n'' + ``measuringUnit='' + measuringUnit + ``\n'' + ``caloriesPerUnit='' + caloriesPerUnit;` The `equals(Object)` should check all the data fileds for each class. For example, two ingredients, Ingredient1 and Ingredient2 are equal if they both have the same name AND measuringUnit AND caloriesPerUnit.

You should also submit an image file UML.jpg with your UML diagram for the homework. You can create it with Violet or any other UML diagram tool.

TestRecipeBook.java class
should contain a main method that tests all the methods in the problem.

Part II: MyProgrammingLab Exercises & Problems (10 points)

Complete all exercises for Chapter Chapter 12 Exception Handling and Text I/O and Chapter 13 Abstract Classes and Interfaces of `www.myprogramminglab.com`.

How to Submit Your Work for Grading

To submit your `.java` files for grading:

1. Login to `http://blackboard.stonybrook.edu` and locate the course account for CSE 114.
2. Click on "Homework Assignments" in the left-hand menu and find the link for this homework assignment.
3. Click on the link for this homework assignment.
4. Click the "Browse My Computer" button and locate the first `.java` file you wish to submit. **Do not submit `.class` files, zip files, rar files, or any other kinds of file except for `.java` files.**
5. Repeat step 4 for each `.java` file you wish to submit.
6. Click the "Submit" button to submit your work for grading.

Oops, I messed up and I need to resubmit a file!

When you submit files to Blackboard, the system assembles them into what's called an "attempt". If you need to resubmit your homework, it is essential that you resubmit ALL files again so that Blackboard assembles them as a single attempt. If you fail to do this, the files you uploaded earlier will not be included in the new "attempt" and might not be graded properly.

Total points: 20 points for the java files and 10 points for MPL (graded separately in Blackboard).