

## Homework 6:

Submit your work (the .java source code files ONLY, not the compiled .class files!) through the “Homework6” link on Brightspace. You may submit an unlimited number of times; we will only grade the last/latest submission attempt, but be sure to attach all of your files to each submission attempt. Be sure to include your name and Stony Brook ID number in a comment at the beginning of each file that you submit.

**Due: Wednesday, November 27, 2024 11:59 PM New York Time.**

**Points: 30 points (5 points for each class plus 5 points for the UML diagram)**

**Submission Instructions: Name your java classes for this assignment as:**

**Files:**

**Ingredient.java,  
RecipeIngredient.java,  
CookingRecipe.java,  
RecipeBook.java,  
TestRecipeBook.java**

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 and 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) returns the RecipeIngredient object corresponding to the ingredient object parameter, 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. These properties should be private, so the class is encapsulated.
- 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 cooking 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 than or equal 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 recipes.

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 fields for each class. For example, two ingredients, Ingredient1 and Ingredient2 are equal if they both have the same name AND measuringUnit AND caloriesPerUnit.
- TestRecipeBook.java class should contain a main method that tests all the methods in the problem.

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