



## Southern New Hampshire University

### IT 511 Stepping Stone Lab Five Guidelines

#### Recipe Class With Accessors and Mutators

**Overview:** Now that you have some experience building basic, single-class applications, here is an opportunity to delve into making a full-fledged application with more than one class. This stepping stone lab is the point where we start to pull all the elements of the course together. You will start to implement a first version of the Recipe class and create a test class to put it through its paces.

Be sure to review the Stepping Stone Lab Five guidelines before beginning this lab. The completed code (.java file) from Stepping Stone Lab Four is included in this module to use as a reference for comparison of your work in Module Five as well as moving forward. This code serves a number of functions:

- It allows you to review your own submitted code.
- You may utilize it as a useful foundation for Stepping Stone Lab Five as well as other assignments to come.
- Together with instructor feedback on your submitted code, you are building your own final project application.

As you are developing the code in this stepping stone lab, you should consider how you can transition it to your final project. Eventually, the user input for collecting the ingredients into an `ArrayList` of strings will be converted to an `ArrayList` of `Ingredient` objects. Think about the variable names you are using and where in the code you are collecting the ingredient input.

SteppingStone5_Recipe
<pre>- recipeName: String - servings: int - recipeIngredients: ArrayList - totalRecipeCalories: double  + getRecipeName(): String + setRecipeName(String): void + getServings(): int + setServings(int): void + getRecipeIngredients(): ArrayList + setRecipeIngredients(ArrayList): void + getTotalRecipeCalories(): double + setTotalRecipeCalories(double): void + printRecipe(): void + addNewRecipe(): SteppingStone5_Recipe</pre>

**Prompt:** In this stepping stone lab assignment, you will build a Recipe class, getting user input to collect the recipe name and serving size, using the ingredient entry code from Stepping Stone Lab Four to add ingredients to the recipe, and calculating the calories per serving. Additionally, you will build your first custom method to print the recipe to the screen.

Specifically, you will need to create the following:

- The instance variables for the class (recipeName, serving size, and recipeIngredients)
- The methods (the accessors/mutators, the constructors, and the extra print details method) for the class
- A custom method to print the recipe out to the console

\*To test the functionality of your finished code, use the SteppingStone5\_RecipeTest.java file

\*Replace the **public static void main(String[] args)** with **public SteppingStone5\_Recipe createNewRecipe()**

**Guidelines for Submission:** This assignment should be submitted as a Java file.

### Extending This Lab for Your Final Project

For your final project:

1. Modify this code to develop a Recipe class:

- A. Change the void main method createNewRecipe() that returns a Recipe

2. **FOR FINAL SUBMISSION ONLY:**

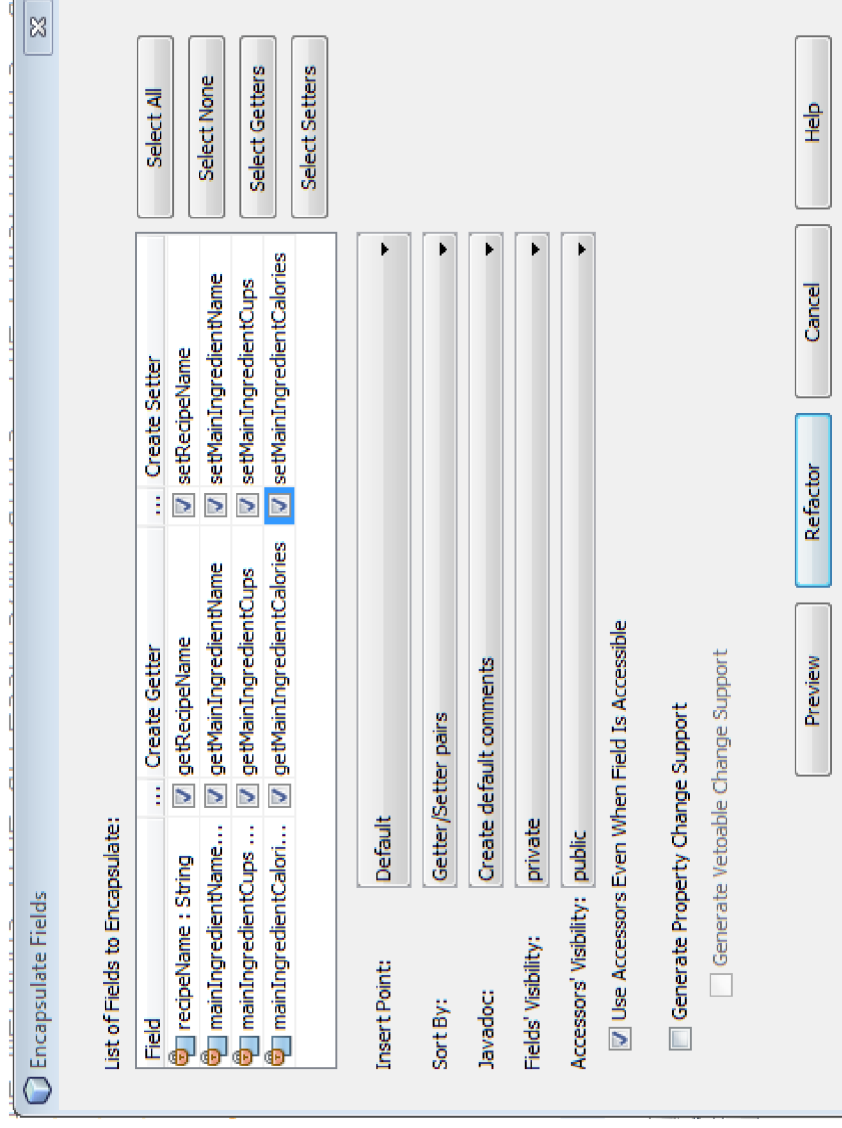
Change the ArrayList type to an Ingredient object. When a user adds an ingredient to the recipe, instead of adding just the ingredient name, you will add the actual ingredient including name, amount, measurement type, and calories. For the Milestone Two submission, the recipeIngredients ArrayList can remain as a string type.

3. Adapt the printRecipe() method to print the amount and measurement type as well as the ingredient name.

4. Create a custom method in the Recipe class. Choose one of the following options:

- A. Print out a recipe with amounts adjusted for a different number of servings.
- B. Create an additional list or ArrayList that allows users to insert step-by-step recipe instructions.
- C. Convert ingredient amounts from English to metric (or vice versa).
- D. Calculate select nutritional information.
- E. Calculate recipe cost.
- F. Propose a suitable alternative to your instructor.

- To add accessors/mutators, choose **Refactor** → **Encapsulate Fields**.



- You can also insert constructors by going to **Source** → **Insert Code**.



- Finally, you can also prompt the user for the recipe information instead of hard-coding it.