# Goal

The goal of this assignment is to build upon the Smoothie Maker used as a learning application in Java 201 and turn it into a full-blown Smoothie Shoppe.

This application introduces features common in real-world applications such as:

* Shopping cart / ordering
* Inventory management
* Data processing

These exercises are based upon a set of Smoothie Recipes and Ingredients provided by your instructor, but at any time please feel free to add your own Ingredients and create your own Recipes.

# Skills Acquired

In this assignment you will learn:

* Eclipse
* Git
* Gradle
* Object-Oriented Design (OOD)
* Object-Oriented Programming (OOP)
* Core Java
* Iterative Development

# The Assignment

## Step 1

Setup Smoothie Shoppe application.

### Success Criteria

* Have a Git repository
* Build from Gradle
* Run from Gradle
* Edit within Eclipse
* Run from Eclipse

### Hints

* <https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository>
* **build.gradle** file provided by instructor
* **.gitignore** file provided by instructor
* <https://docs.gradle.org/current/userguide/tutorial_gradle_command_line.html>
* <https://docs.gradle.org/current/userguide/eclipse_plugin.html>
* <http://projects.eclipse.org/projects/tools.buildship/downloads>

### Instructions

1. Copy the Java201 project into a new folder within the Eclipse workspace
2. Add the **.gitignore** file to the project folder
3. Create a Git repository and make an initial commit
4. Add the **build.gradle** file to the project
5. Reorganize the files to conform to the Gradle expectations:  
   Source: /src/main/java  
   Resources: /src/main/resources
6. Install the Buildship plugin for Eclipse to add Gradle support
7. Commit all changes to git and create a **step1** tag

### Verification

Executing **gradle run** should produce output similar to the following:

:compileJava UP-TO-DATE

:processResources UP-TO-DATE

:classes UP-TO-DATE

:run

207 [main] INFO org.elevenfifty.smoothie.Main - Ingredients:

FRUIT Orange

FRUIT Banana

207 [main] INFO org.elevenfifty.smoothie.Main - Instructions:

Peel FRUIT:Banana

Cut FRUIT:Banana

Add FRUIT:Banana

Peel FRUIT:Orange

Cut FRUIT:Orange

Add FRUIT:Orange

BUILD SUCCESSFUL

Total time: 2.582 secs

## Step 2

Read Ingredients and Recipes from file.

### Success Criteria

* Load Ingredients from file
* Load Recipes from file
* Output remains the same

### Hints

* Files **recipes.csv** and **ingredients.csv** provided by instructor
* <https://www.mkyong.com/java/how-to-read-and-parse-csv-file-in-java/>
* <https://www.mkyong.com/java/java-read-a-file-from-resources-folder/>
* Consider using a **Map** to lookup Recipes by name

### Instructions

1. Add the CSV files to the resources folder
2. Create a Reader for CSV files
3. Create a new application configuration which loads Recipes and Ingredients from file
4. Remove all XML files
5. Remove all references to Spring
6. Modify the Main application class to work with the new configuration
7. Commit all changes to git and create a **step2** tag

### Verification

Executing **gradle run** should produce output similar to that from step 2, minus the Spring overhead, something like:

:compileJava

:processResources

:classes

:run

Ingredients:

FRUIT Orange

FRUIT Banana

Instructions:

Peel FRUIT:Banana

Cut FRUIT:Banana

Add FRUIT:Banana

Peel FRUIT:Orange

Cut FRUIT:Orange

Add FRUIT:Orange

0.6

BUILD SUCCESSFUL

Total time: 3.062 secs

## Step 3

Browse smoothie recipes and allow one to be selected.

### Success Criteria

* Print indexed list of Recipes
* Prompt for input
* Display details for selected Recipe

### Hints

* Consider using a **List** to keep Recipes in a consistent order
* <https://www.mkyong.com/java/how-to-read-input-from-console-java/>
* Consider using **System.out.format** to print the Recipe list

### Instructions

1. Create utility class to print Recipes and Smoothies (consistent with format used within **Main.main** and **Main.printPretty**)
2. Create a Recipe browser to display all available Recipes
3. Print Recipe index (1-N), Name, and Cost (formatted as currency)
4. Prompt for input to select a Recipe to print (should accept the recipe index # as input)
5. Print Recipe details
6. Commit all changes to git and create a **step3** tag

### Verification

Executing **gradle run -q** should produce output similar to the following:

1: Best Smoothie $1.10

2: Another Smoothie $2.10

3: Green Smoothie $0.90

Select a Smoothie Recipe: 2

Another Smoothie

Ingredients:

FRUIT Mango

FRUIT Watermelon

FRUIT Guava

BASE Water

MISC Ice

Instructions:

Peel FRUIT:Mango

Pit FRUIT:Mango

Cut FRUIT:Mango

Add FRUIT:Mango

Peel FRUIT:Watermelon

Cut FRUIT:Watermelon

Add FRUIT:Watermelon

Peel FRUIT:Guava

Pit FRUIT:Guava

Cut FRUIT:Guava

Add FRUIT:Guava

Add BASE:Water

Add MISC:Ice

BLEND!

Enjoy!

2.1

## Step 4

Add Inventory (quantity on hand) to the application.

### Success Criteria

* Add Quantity field to Ingredients
* Ready quantity from file
* Display quantity on the screen
* Subtract quantity when Recipe is printed

### Hints

* Consider **String.format** for displaying quantity
* Review **Ingredient.toString**

### Instructions

1. Add Quantity column to CSV files
2. Set all quantities to a number between 1 - 10
3. Add quantity field to Ingredient class
4. Read quantity value from file
5. Display quantity whenever Ingredient is displayed
6. Decrement quantity whenever Ingredient is displayed
7. Rename Main class to **SmoothieShoppe**
8. Commit all changes to git and create a **step4** tag

### Verification

Executing **gradle run -q** should produce output similar to the following:

1: Best Smoothie $1.10

2: Another Smoothie $2.10

3: Green Smoothie $0.90

Select a Smoothie Recipe: 1

Best Smoothie

Ingredients:

FRUIT Orange (1 on hand)

FRUIT Banana (4 on hand)

FRUIT Strawberry (4 on hand)

BASE Milk (4 on hand)

MISC Ice (4 on hand)

Instructions:

Peel FRUIT:Orange

Cut FRUIT:Orange

Add FRUIT:Orange

Peel FRUIT:Banana

Cut FRUIT:Banana

Add FRUIT:Banana

Wash FRUIT:Strawberry

Cut FRUIT:Strawberry

Add FRUIT:Strawberry

Add BASE:Milk

Add MISC:Ice

BLEND!

Enjoy!

1.1

## Step 5

### Success Criteria

* App runs in a loop
* Type ‘q’ to quit
* Recipes with ingredients of insufficient quantity display “Insufficient Inventory”
* Unable to select recipes with insufficient quantity of one or more ingredients

### Hints

* The program loop is similar to a Game Loop: <http://gameprogrammingpatterns.com/game-loop.html>
* If you get stuck you can use CTRL+C to break out of an infinite loop

### Instructions

1. Add handling for ‘q’ input to quit the application
2. Put the recipe prompt inside an infinite loop
3. Create a test for available quantity (>0) for each ingredient within the recipe
4. Display a message for recipes with insufficient quantity instead of creating the smoothie
5. Commit all changes to git and create a **step5** tag

### Verification

Executing **gradle run -q** should produce output similar to the following:

Available Recipes:

1: Best Smoothie $1.10

2: Another Smoothie $2.10

3: Green Smoothie $0.90

Select a Smoothie Recipe (q to quit): 2

Another Smoothie

Ingredients:

FRUIT Mango (4 on hand)

FRUIT Watermelon (2 on hand)

FRUIT Guava (0 on hand)

BASE Water (6 on hand)

MISC Ice (3 on hand)

Instructions:

Peel FRUIT:Mango

Pit FRUIT:Mango

Cut FRUIT:Mango

Add FRUIT:Mango

Peel FRUIT:Watermelon

Cut FRUIT:Watermelon

Add FRUIT:Watermelon

Peel FRUIT:Guava

Pit FRUIT:Guava

Cut FRUIT:Guava

Add FRUIT:Guava

Add BASE:Water

Add MISC:Ice

BLEND!

Enjoy!

Cost: $2.10

Available Recipes:

1: Best Smoothie $1.10

2: Another Smoothie (insufficient inventory)

3: Green Smoothie $0.90

Select a Smoothie Recipe (q to quit): 2

Insufficient Inventory

Available Recipes:

1: Best Smoothie $1.10

2: Another Smoothie (insufficient inventory)

3: Green Smoothie $0.90

Select a Smoothie Recipe (q to quit): q

## Step 6

### Success Criteria

* Application menu offers options to Browse, Make Smoothies, or Quit
* Selecting a smoothie adds it to a list
* Process saved smoothies all at once, displaying each one in the order it was selected

### Hints

* Consider adding the selected Recipe to a **List** when selected
* Subtracting ingredients when added (rather than when processed) will prevent selecting Smoothies that cannot be made

### Instructions

1. Create application menu
2. Selecting smoothies adds them to a list (instead of printing immediately)
3. Display list of selected smoothies each time a new one is selected
4. Commit all changes to git and create a **step6** tag

### Verification

Executing **gradle run -q** should produce output similar to the following:

\*\* Smoothie Shoppe \*\*

Application Menu:

[b] Browse Smoothies

[m] Make Smoothies

[q] Quit

Select a Smoothie Recipe (q to quit): m

No smoothies selected!

Application Menu:

[b] Browse Smoothies

[m] Make Smoothies

[q] Quit

Select a Smoothie Recipe (q to quit): b

Available Recipes:

1: Best Smoothie $1.10

2: Another Smoothie $2.10

3: Green Smoothie $0.90

Select a Smoothie Recipe (q to quit): 1

Selected Smoothies:

Best Smoothie

Application Menu:

[b] Browse Smoothies

[m] Make Smoothies

[q] Quit

Select a Smoothie Recipe (q to quit): b

Available Recipes:

1: Best Smoothie $1.10

2: Another Smoothie $2.10

3: Green Smoothie $0.90

Select a Smoothie Recipe (q to quit): 2

Selected Smoothies:

Best Smoothie

Another Smoothie

Application Menu:

[b] Browse Smoothies

[m] Make Smoothies

[q] Quit

Select a Smoothie Recipe (q to quit): m

Best Smoothie

Ingredients:

FRUIT Orange (3 on hand)

FRUIT Banana (6 on hand)

FRUIT Strawberry (5 on hand)

BASE Milk (5 on hand)

MISC Ice (4 on hand)

Instructions:

Peel FRUIT:Orange

Cut FRUIT:Orange

Add FRUIT:Orange

Peel FRUIT:Banana

Cut FRUIT:Banana

Add FRUIT:Banana

Wash FRUIT:Strawberry

Cut FRUIT:Strawberry

Add FRUIT:Strawberry

Add BASE:Milk

Add MISC:Ice

BLEND!

Enjoy!

Cost: $1.10

Another Smoothie

Ingredients:

FRUIT Mango (6 on hand)

FRUIT Watermelon (4 on hand)

FRUIT Guava (2 on hand)

BASE Water (8 on hand)

MISC Ice (4 on hand)

Instructions:

Peel FRUIT:Mango

Pit FRUIT:Mango

Cut FRUIT:Mango

Add FRUIT:Mango

Peel FRUIT:Watermelon

Cut FRUIT:Watermelon

Add FRUIT:Watermelon

Peel FRUIT:Guava

Pit FRUIT:Guava

Cut FRUIT:Guava

Add FRUIT:Guava

Add BASE:Water

Add MISC:Ice

BLEND!

Enjoy!

Cost: $2.10

Application Menu:

[b] Browse Smoothies

[m] Make Smoothies

[q] Quit

Select a Smoothie Recipe (q to quit): q

# Challenges

## Bronze

To complete the Bronze Challenge your app should begin looking a bit more polished. Errors should show messages instead of stack traces, and all output should be nicely and consistently formatted for readability. You should also add a Shopping Cart, which groups items by name, lists quantity in the cart, and allows removing items from the cart.

### Success Criteria

* Handle unexpected input gracefully with meaningful error messages and no stack traces
* Ensure all output is consistent, and formatted nicely for readability
* Add new option to application menu “I” to list available Ingredients with their current quantity  
  - Should list actual quantity available, unmodified by any pending orders
* Add new option to application menu “C” to view cart  
  - From here options include “R” Return (to the application menu) and “M” Make Smoothies  
  - Selecting a Recipe by # should remove it from the Cart
* Cart should group Smoothies by Name and show Order Quantity

## Silver

To complete the Silver Challenge your app will need to list at least 30 Smoothie Recipes, displayed in pages of 10 at a time. Navigation will be N = Next, P = Previous, 1-0 selects the displayed smoothie. Once added to the cart, the cart itself will need to allow changes in quantity. Add Units for all ingredients. This may be something like “2 ice cubes”, “1 medium banana”, “1 cup water”, etc.

### Success Criteria

* Adding a Smoothie to the Cart takes you directly to the Cart  
  - Selecting Return should return you to the exact page you had previously been browsing
* Add Pagination feature to Browsing for smoothies  
  - N = Next Page  
  - P = Previous Page  
  - 1-0 = Add Smoothie to Cart  
  - Create at least 30 new Smoothie Recipes to test the functionality
* Modify Shopping Cart to allow Quantity to be updated  
  - Instead of immediately removing an item, selecting an item should display a menu  
  - Menu should list “R” to remove item, or enter a number for the new quantity
* Update **ingredients.csv** to include the new units  
  - Display units anywhere quantities are displayed  
  - Take singular/plural naming into consideration (1 ice cube, 2 ice cubes)

## Gold

To complete the Gold Challenge you need to add Administration capabilities to your application. This should allow you to browse, edit, remove, and create both Ingredients and Recipes. All changes must be saved back to the **ingredients.csv** and **recipes.csv** files.

### Success Criteria

* Update main application menu  
  - Remove “I” for inventory option  
  - Add “A” for Administrative Console
* Create Admin menu  
  - “R” to list recipes  
  - “I” to list ingredients  
  - “N” for new recipe  
  - “A” to add ingredient  
  - “X” to exit Admin console
* When a Recipe or Ingredient is selected from the list, prompt for changes to each field  
  - Pressing enter with no changes keeps the field unchanged  
  - Allow Ingredient inventory to be increased
* Persist all changes to ingredients and recipes to the appropriate CSV file

## Platinum

To complete the Platinum Challenge you will refine the Checkout process to make the selection of Smoothies feel more like an online store. The Cart will need to list Smoothies, Quantity Ordered, Price, and Subtotal (Price x Quantity) along with a final Total at the bottom. This should be nicely formatted in columns. When checking out, the output from creating Smoothies should be consolidated - Each different fruit and vegetable should only be peeled, pitted, or cut one time for all smoothies (must aggregate ingredients for all smoothies). Then list the steps for actually mixing together the smoothies individually, making all smoothies of one type at a time.

### Success Criteria

* Update Cart page to include Price, Subtotal, and Total
* Format Cart page to line up numbers into columns
* Each Fruit and Vegetable are only processed one time, regardless of number of smoothies
* Smoothies of the same type are blended at once