Brewery Control System - Scoring as per Metrics

1) Weighted Methods per Class

- Main Class

Attributes: 0

Methods: 1

- ❖ main - Entry point of the system, interacting with all subsystems.
- ❖ No. of lines in the method: 122
- ❖ Cyclomatic complexity: Moderate (loops, conditions)

Total: 0 + 122 = 122

- InventorySystem Class

Attributes: 2

Methods: 7

- ❖ Name of the methods: addIngredient, increaseIngredientQuantity, verifyQuantity, reduceIngredient, ingredientExists, findIngredientByName, displayIngredients
- ❖ No. of lines in methods: 78
- ❖ Cyclomatic complexity: Low (simple operations per method)

Total: 2 + 78 = 80

- ProductionSystem Class

Attributes: 3

Methods: 3

- ❖ Name of the methods: createBatch, printContainers, displayContainers
- ❖ No. of lines in methods: 95
- ❖ Cyclomatic complexity: High (complex operations with multiple objects)

Total: 3 + 95 = 98

- RecipeLibrary Class

Attributes: 1

Methods: 4

- Name of the methods: addAllRecipe, getRecipe, showAllRecipes, startDefaultRecipes
- No. of lines in methods: 44
- Cyclomatic complexity: Low (fetch and store operations)

Total: 1 + 44 = 45

- Ingredient Class (InventorySystem)

Attributes: 2

Methods: 3

- Name of the methods: getName, getQuantity, setQuantity
- No. of lines in methods: 12
- Cyclomatic complexity: Low (getter/setter methods)

Total: 2 + 12 = 14

- Ingredient Class (RecipeLibrary)

Attributes: 2

Methods: 2

- Name of the methods: getName, getQuantity
- No. of lines in methods: 7
- Cyclomatic complexity: Low

Total: 2 + 7 = 9

- Recipe Class

Attributes: 2

Methods: 3

- Name of the methods: addIngredient, getName, getAllIngredients
- No. of lines in methods: 24
- Cyclomatic complexity: Low

Total: 2 + 24 = 26

- CustomList Class

Attributes: 2

Methods: 4

- ❖ Name of the methods: add, resize, get, size
- ❖ No. of lines in methods: 22
- ❖ Cyclomatic complexity: Low (dynamic array operations)

Total: 2 + 22 = 24

---

2) Depth of Inheritance Tree

Explanation: None of the classes in this program use inheritance. Thus, the score is 0 for Depth of Inheritance Tree for all classes.

- All classes: No inheritance

---

3) Number of Children

Explanation: None of the classes in the system have child classes. The score is 0 for Number of Children for all classes.

- All classes: 0

---

4) Coupling between Objects

Explanation: The coupling between objects in the system is moderate. Some classes depend on others to function, but there is no excessive interdependency.

- Main Class is tightly coupled with InventorySystem, ProductionSystem, and RecipeLibrary.

- InventorySystem and RecipeLibrary have minimal coupling.

---

5) Response for a Class (RFC)

Explanation: Here, we calculate how many methods each class calls from different classes.

- Main Class: RFC = 3

o Calls methods from InventorySystem, ProductionSystem, and RecipeLibrary.

- InventorySystem Class: RFC = 1

o Calls Ingredient-related methods.

- ProductionSystem Class: RFC = 4

o Calls methods from Recipe, Ingredient, and InventorySystem.

- RecipeLibrary Class: RFC = 1

o Calls methods from Recipe.

- Ingredient Class (InventorySystem): RFC = 0

o No method calls from other classes.

- Ingredient Class (RecipeLibrary): RFC = 0

o No method calls from other classes.

- Recipe Class: RFC = 1

o Calls methods from Ingredient.

- CustomList Class: RFC = 0

o No method calls from other classes.

Total RFC in the system: 10

---

6) Cohesion Across Methods

Explanation: Most classes in the system show high cohesion, as their methods are closely related to their core functionality.

- Main Class: High cohesion (manages system interaction)

- InventorySystem Class: High cohesion (manages ingredients)

- ProductionSystem Class: High cohesion (manages production of cheesecakes)

- RecipeLibrary Class: High cohesion (manages recipes)

- Ingredient Class (InventorySystem): High cohesion (getter/setter operations)

- Ingredient Class (RecipeLibrary): High cohesion (getter/setter operations)

- Recipe Class: High cohesion (manages recipe creation)

- CustomList Class: High cohesion (manages dynamic list operations)

Total score after completing the whole program is:
122+80+98+45+14+9+26+24+10
= 428