

Link to repo: <https://github.com/JakeHresh/ser316-spring2021-C-jhreshch/tree/metrics2>

## Task 1.

PMD report			
Problems found			
#	File	Line	Problem
1	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	10	<a href="#">Avoid using implementation types like 'LinkedList'; use the interface instead</a>
2	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	11	<a href="#">Avoid using implementation types like 'LinkedList'; use the interface instead</a>
3	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	41	<a href="#">The method 'addNoise(NoiseMaker)' has a cyclomatic complexity of 4.</a>
4	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	6	<a href="#">Avoid really long classes.</a>
5	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	37	<a href="#">The method 'getCost(Bear)' has a cyclomatic complexity of 5.</a>
6	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	37	<a href="#">The method 'getCost(Bear)' has an NPath complexity of 12</a>
7	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	67	<a href="#">The method 'getRawCost(Bear)' has a cyclomatic complexity of 4.</a>
8	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	97	<a href="#">The method 'calculateTax()' has a cyclomatic complexity of 6.</a>
9	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	129	<a href="#">The method 'addBear(Bear)' has a cyclomatic complexity of 2.</a>
10	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	140	<a href="#">The method 'removeBear(Bear)' has a cyclomatic complexity of 2.</a>
11	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	157	<a href="#">Avoid really long methods.</a>
12	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	157	<a href="#">The method 'checkout()' has a cyclomatic complexity of 11.</a>
13	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	157	<a href="#">The method 'checkout()' has an NPath complexity of 240</a>
14	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Clothing.java	17	<a href="#">The method 'compareTo(Clothing)' has a cyclomatic complexity of 3.</a>
15	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>Main.java	7	<a href="#">Avoid really long methods.</a>
16	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	14	<a href="#">The constructor 'NoiseMaker(String, String, Location)' has a cyclomatic complexity of 2.</a>
17	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	29	<a href="#">Avoid really long methods.</a>
18	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	29	<a href="#">The constructor 'NoiseMaker(String, String, int)' has a cyclomatic complexity of 5.</a>
19	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Stuffing.java	13	<a href="#">The constructor 'Stuffing(stuffing)' has a cyclomatic complexity of 4.</a>

There are a couple of problems relating to loose coupling where there is the use of an implementation of something like a LinkedList, which limits the flexibility of refining the implementation as requirements change. The fix would be to use the interface that LinkedList implements instead. These errors, 1 and 2, map to `<rule ref="category/java/bestpractices.xml/LooseCoupling" />`.

Errors 3, 5, 7, 8, 9, 10, 12, 14, 16, 18, and 19 relate to the cyclomatic complexity of methods, which relates to the number of decision points contained in a method plus one for the method entry.

Complexities of 1-4 are low, complexities of 5-7 are moderate, complexities of 8-10 are high, and complexities of 11+ are very high. Most of the cyclomatic complexities are low to moderate. However, checkout() has a complexity of 11, which might be too high. These errors map to `<rule ref="category/java/design.xml/CyclomaticComplexity">`

Error 4 relates to long class length, which is measured in lines of code. The minimum class length required for my BearWorkshop.java class is 100 LOC, which might be too small. 225 LOC is still a

reasonable class length, especially when the class contains several comments. The error maps to `<rule ref="category/java/design.xml/ExcessiveClassLength">`

Errors 11, 15, and 29 relate to long method length, which is also measured in lines of code. The minimum method length required for my methods is 30 LOC, which also might be too small. Some methods include the main method, which might reasonably contain more than 30 lines for testing purposes. The error maps to `<rule ref="category/java/design.xml/ExcessiveMethodLength">`

Error 13 relates to NPath complexity, which relates to the number of execution paths in a method.

According to the documentation, a threshold of 200 is the point where complexity reduction should take place to reduce complexity and increase readability, which means that the threshold provided in the xml must be increased from 10 to 200. The checkout() method has a complexity of 240, which means that time should probably be spent reducing that complexity. This error maps to `<rule ref="category/java/design.xml/NPathComplexity">`

## Task 2.

### PMD report

#### Problems found

#	File	Line	Problem
1	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	9	<a href="#">Avoid really long classes.</a>
2	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	13	<a href="#">Avoid using implementation types like 'LinkedList'; use the interface instead</a>
3	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	14	<a href="#">Avoid using implementation types like 'LinkedList'; use the interface instead</a>
4	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	59	<a href="#">The method 'addNoise(NoiseMaker)' has a cyclomatic complexity of 4.</a>
5	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	92	<a href="#">The method 'equals(Object)' has a cyclomatic complexity of 5.</a>
6	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	92	<a href="#">The method 'equals(Object)' has an NPath complexity of 16</a>
7	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	24	<a href="#">Avoid really long classes.</a>
8	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	55	<a href="#">The method 'getCost(Bear)' has a cyclomatic complexity of 5.</a>
9	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	55	<a href="#">The method 'getCost(Bear)' has an NPath complexity of 12</a>
10	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	87	<a href="#">The method 'getRawCost(Bear)' has a cyclomatic complexity of 4.</a>
11	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	120	<a href="#">The method 'calculateTax()' has a cyclomatic complexity of 6.</a>
12	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	159	<a href="#">The method 'removeBear(Bear)' has a cyclomatic complexity of 2.</a>
13	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	185	<a href="#">Avoid really long methods.</a>
14	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	185	<a href="#">The method 'checkout()' has a cyclomatic complexity of 4.</a>
15	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	246	<a href="#">Avoid really long methods.</a>
16	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	246	<a href="#">The method 'calculateSavings()' has a cyclomatic complexity of 8.</a>
17	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	246	<a href="#">The method 'calculateSavings()' has an NPath complexity of 50</a>
18	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Clothing.java	41	<a href="#">The method 'equals(Object)' has a cyclomatic complexity of 5.</a>
19	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Clothing.java	41	<a href="#">The method 'equals(Object)' has an NPath complexity of 16</a>
20	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>Main.java	11	<a href="#">Avoid really long methods.</a>
21	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	17	<a href="#">The constructor 'NoiseMaker(String, String, Location)' has a cyclomatic complexity of 2.</a>
22	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	35	<a href="#">The constructor 'NoiseMaker(String, String, int)' has a cyclomatic complexity of 5.</a>
23	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>Stuffing.java	17	<a href="#">The constructor 'Stuffing(StuffingE)' has a cyclomatic complexity of 4.</a>

The added work introduced a few more PMD errors in increasing the lengths of methods, like with the corrected implementation of `calculateSavings()`. The complexity of `checkout()` was also reduced to 4.

Otherwise, the other problems still persist since no changes were made directly to those methods and classes.

### Task 3.1

PMD report				
Problems found				
#	File	Line	Problem	
1	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	9	<a href="#">Avoid really long classes.</a>	
2	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	13	<a href="#">Avoid using implementation types like 'LinkedList', use the interface instead</a>	
3	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	14	<a href="#">Avoid using implementation types like 'LinkedList', use the interface instead</a>	
4	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	59	<a href="#">The method 'addNoise(NoiseMaker)' has a cyclomatic complexity of 4.</a>	
5	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	92	<a href="#">The method 'equals(Object)' has a cyclomatic complexity of 5.</a>	
6	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Bear.java	92	<a href="#">The method 'equals(Object)' has an NPath complexity of 16</a>	
7	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	24	<a href="#">Avoid really long classes.</a>	
8	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	58	<a href="#">The method 'getCost(Bear)' has a cyclomatic complexity of 5.</a>	
9	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	58	<a href="#">The method 'getCost(Bear)' has an NPath complexity of 12</a>	
10	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	90	<a href="#">The method 'getRawCost(Bear)' has a cyclomatic complexity of 4.</a>	
11	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	163	<a href="#">The method 'removeBear(Bear)' has a cyclomatic complexity of 2.</a>	
12	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	189	<a href="#">The method 'checkout()' has a cyclomatic complexity of 4.</a>	
13	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	209	<a href="#">Avoid really long methods.</a>	
14	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	209	<a href="#">The method 'calculateSavings()' has a cyclomatic complexity of 8.</a>	
15	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\BearWorkshop.java	209	<a href="#">The method 'calculateSavings()' has an NPath complexity of 50</a>	
16	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Clothing.java	41	<a href="#">The method 'equals(Object)' has a cyclomatic complexity of 5.</a>	
17	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java\Clothing.java	41	<a href="#">The method 'equals(Object)' has an NPath complexity of 16</a>	
18	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>Main.java	11	<a href="#">Avoid really long methods.</a>	
19	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	17	<a href="#">The constructor 'NoiseMaker(String, String, Location)' has a cyclomatic complexity of 2.</a>	
20	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>NoiseMaker.java	35	<a href="#">The constructor 'NoiseMaker(String, String, int)' has a cyclomatic complexity of 5.</a>	
21	D:\SER316NewClonedAssignmentRepo\ser316-spring2021-C-jhreshch\src\main\java>Stuffing.java	17	<a href="#">The constructor 'Stuffing(StuffingE)' has a cyclomatic complexity of 4.</a>	

The `getTax()` method is here called `calculateTax()` and exists in the `BearWorkshop.java` class. Because there were a set of if-else conditions that increased the cyclomatic complexity of the function, I decided to include a tax attribute to the class and modified the existing constructors so that one can pass in the tax value as a parameter, thereby removing any decision making on the method and placing that decision point on the user. This required some slight updating of unit tests. However, error 11 no longer points to the line where `calculateTax()` method exists.

### Task 3.2

On reviewing the `calculateSavings()` method, I determined that there is no duplicate code in that method. The method works by creating an array of doubles, representing the prices-to-be of each bear

in the shopping cart. Then, for each bear in the shopping cart, there will be a tracker keeping track of the number of non-free accessories. For each bear, there are two loops, one that calculates the prices of the non-free clothes, and the other that calculates the prices of noisemaker accessories. After these prices are calculated with the ink price and casing price, the accessories of each bear are counted within the main loop iterating over the bear collection and used to add to the savings value and reduce the individual bear's price by that calculated savings. Once this is done for each bear, a different loop determines the collection of free bears and adds to the savings the prices of each free bear. The result is finally returned. While there are many decision points in this function, the presence of such iterations prevents duplicate code from appearing. Thus, nothing needs to be changed.

#### Task 5 (sic.)

I was denied permission to view Chapter 3 of the Fowler reading on code smells. However, there are code smells in the program that can be checked.

I included the following checks:

- GodClass – for overly large and responsible classes. This will assist in identifying areas where large classes exist and will encourage not just the shrinking of classes, but also the distribution of functionality across other smaller classes.
- ExcessiveParameterList – for methods that contain too many parameters, making them difficult to maintain. This will assist in identifying areas where methods contain too many parameters.
- AvoidCatchingGenericException – prevents catching of NullPointerException, RuntimeException, and Exception. This might address speculative generality, which tries to anticipate making the program flexible for the future. It would be more useful to catch exceptions that are bound to result.

- DataClass – for the data class code smell, which is defined by classes that act only as simple data holders that lack functionality. This results in “poor data-behaviour proximity”.

#### Task 6 (sic.)

Generally speaking, because this program is quite small, refactoring to fix code smells doesn't make much sense. While there are plenty of code smells that can be identified in the program through PMD, any benefits in refactoring at this point would be marginal. Once the program becomes larger, it might make sense to do some refactoring so that there can be larger maintenance gains in the future.

The primary identified code smells are long classes (Bear.java, BearWorkshop.java), long methods (BearWorkshop.calculateSavings(), Main.main()), and data class (Casing.java, Customer.java, Embroider.java, NoiseMaker.java, Stuffing.java). The fact that many of these classes might reflect the data class smell reflects the simplicity of this program. Thus, in a sense, some refactoring is needed by making these classes more functional, thereby making the program more robust and filled with features. However, this is less a matter of refactoring and more a matter of adding features to the system.