

# Assignment 2 - Unit Tests

nswecker SER 316

## Reading the given code (10 points)

DONE

## Five things that I would change

1. Creating a bear requires first designating the stuffing. I think bear creation should happen without being dependent on stuffing type.
2. Adding a noise maker looks confusing to me. Having a default setting that puts the price at \$5 may cause future problems. Specify the exact cost for the exact location.
3. There are no getters or setters for the Customer class. Its hard to test customer if I have no access to their attributes.
4. The compareTo method in Clothing has no description. I have no idea what this method is for, and so have no basis for how to test it.
5. checkout() for BearWorkshop indicates an age restriction on purchasing. I have no idea why this exists.

## Task 1

10. Which implementation adheres to the specification best?

Answer: implementation 1, 3, and 4 past 15 out of 17 tests, while 0 and 2 past 14 out of 17. Obviously none passed all tests.

11. List errors found

- a. app does not properly calculate the 10% discount for 10+ paid accessories
- b. app does not always calculate the correct ink costs for embroidery
- c. app does not always calcute the free ink discount for \$70+ bears

The screenshot shows an IDE with two main windows. The left window displays the source code for a project, including files like `Customer.java`, `Bear.java`, `BearWorkshop.java`, and `Clothing.java`. The right window shows the test results for the `main.java` file. The test results table is as follows:

Test	Passed	Failed	Skipped	Ignored	Not Executed	Not Run	Not Executed	Not Run
Test 1	100%	0%	0%	0%	0%	0%	0%	0%
Test 2	100%	0%	0%	0%	0%	0%	0%	0%
Test 3	100%	0%	0%	0%	0%	0%	0%	0%
Test 4	100%	0%	0%	0%	0%	0%	0%	0%
Test 5	100%	0%	0%	0%	0%	0%	0%	0%
Test 6	100%	0%	0%	0%	0%	0%	0%	0%
Test 7	100%	0%	0%	0%	0%	0%	0%	0%
Test 8	100%	0%	0%	0%	0%	0%	0%	0%
Test 9	100%	0%	0%	0%	0%	0%	0%	0%
Test 10	100%	0%	0%	0%	0%	0%	0%	0%
Test 11	100%	0%	0%	0%	0%	0%	0%	0%
Test 12	100%	0%	0%	0%	0%	0%	0%	0%
Test 13	100%	0%	0%	0%	0%	0%	0%	0%
Test 14	100%	0%	0%	0%	0%	0%	0%	0%
Test 15	100%	0%	0%	0%	0%	0%	0%	0%
Test 16	100%	0%	0%	0%	0%	0%	0%	0%
Test 17	100%	0%	0%	0%	0%	0%	0%	0%
Total	100%	0%	0%	0%	0%	0%	0%	0%

The right window also shows a detailed view of the test results, including a list of test cases and their outcomes. The test cases are listed in a table with columns for the test name, the expected result, and the actual result. The test cases are as follows:

Test Case	Expected Result	Actual Result
Test 1	100%	100%
Test 2	100%	100%
Test 3	100%	100%
Test 4	100%	100%
Test 5	100%	100%
Test 6	100%	100%
Test 7	100%	100%
Test 8	100%	100%
Test 9	100%	100%
Test 10	100%	100%
Test 11	100%	100%
Test 12	100%	100%
Test 13	100%	100%
Test 14	100%	100%
Test 15	100%	100%
Test 16	100%	100%
Test 17	100%	100%

## Task 2

