**CMSC203 Assignment 6 Implementation (Documentation)**

Class: CMSC203 CRN XXXX

 Program: Assignment # 6

Instructor: Professor Grinberg

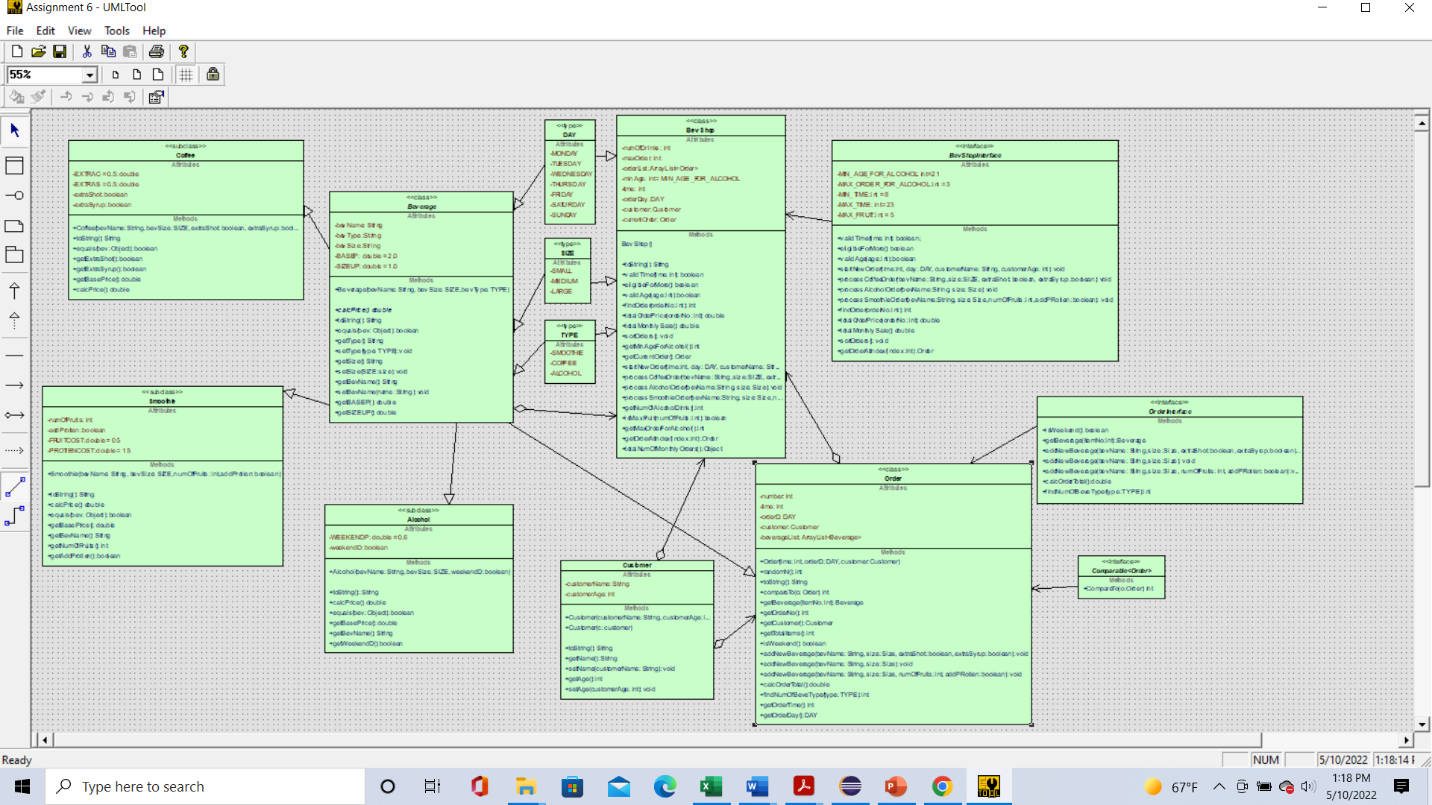
Summary of Description: The BevShop offers 3 types of beverages in 3 different sizes. All the beverage types have a base price and have additional charges depending on the size and specific add-ons for each type of beverage. The BevShop program functions by creating and processing orders of different types of beverages, providing information on all the orders, getting the total amount on a specific order, getting the monthly total number of orders, and getting the monthly sale report.

 Due Date: 05/02/2022

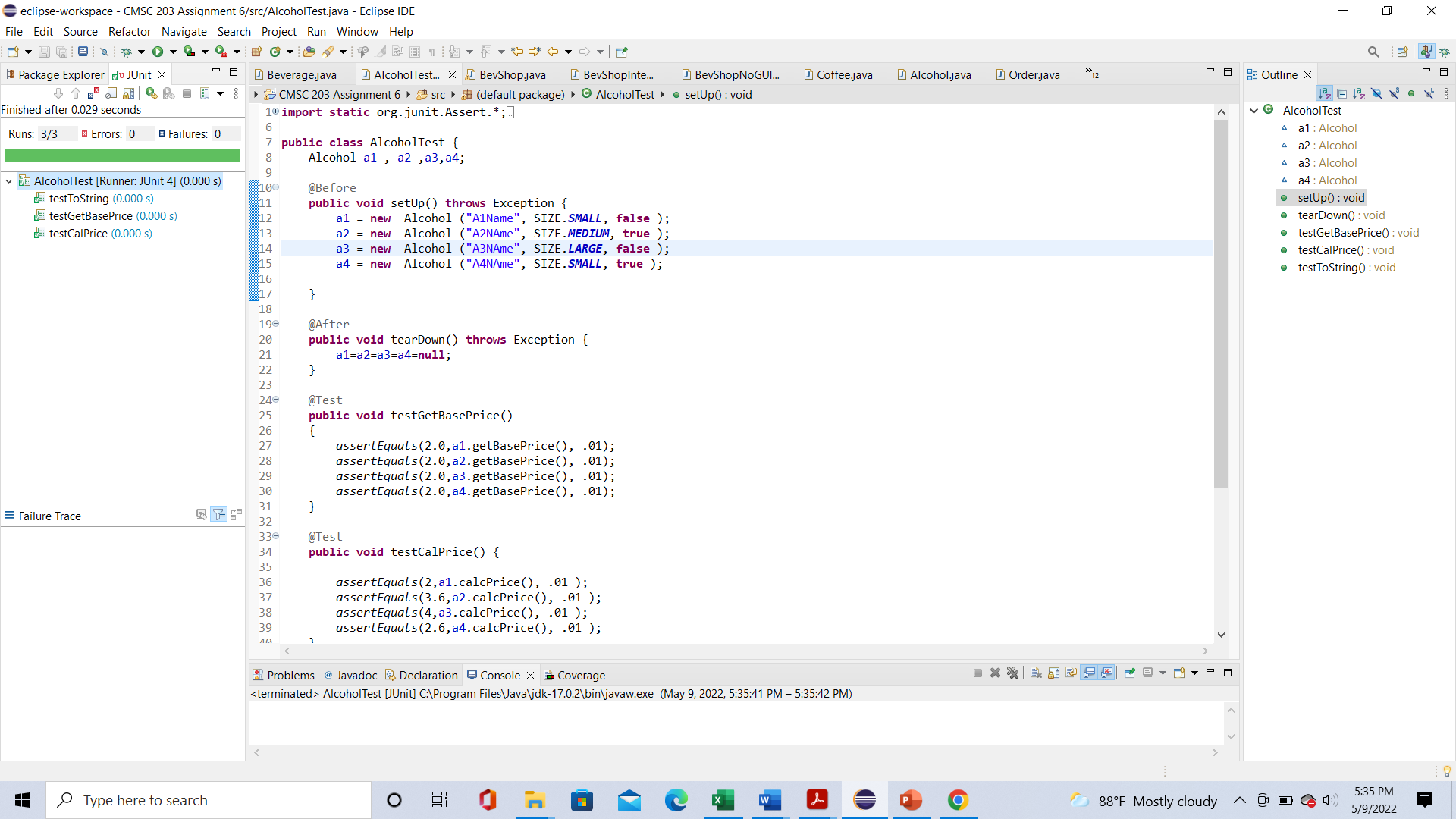
 Integrity Pledge: I pledge that I have completed the programming assignment independently.

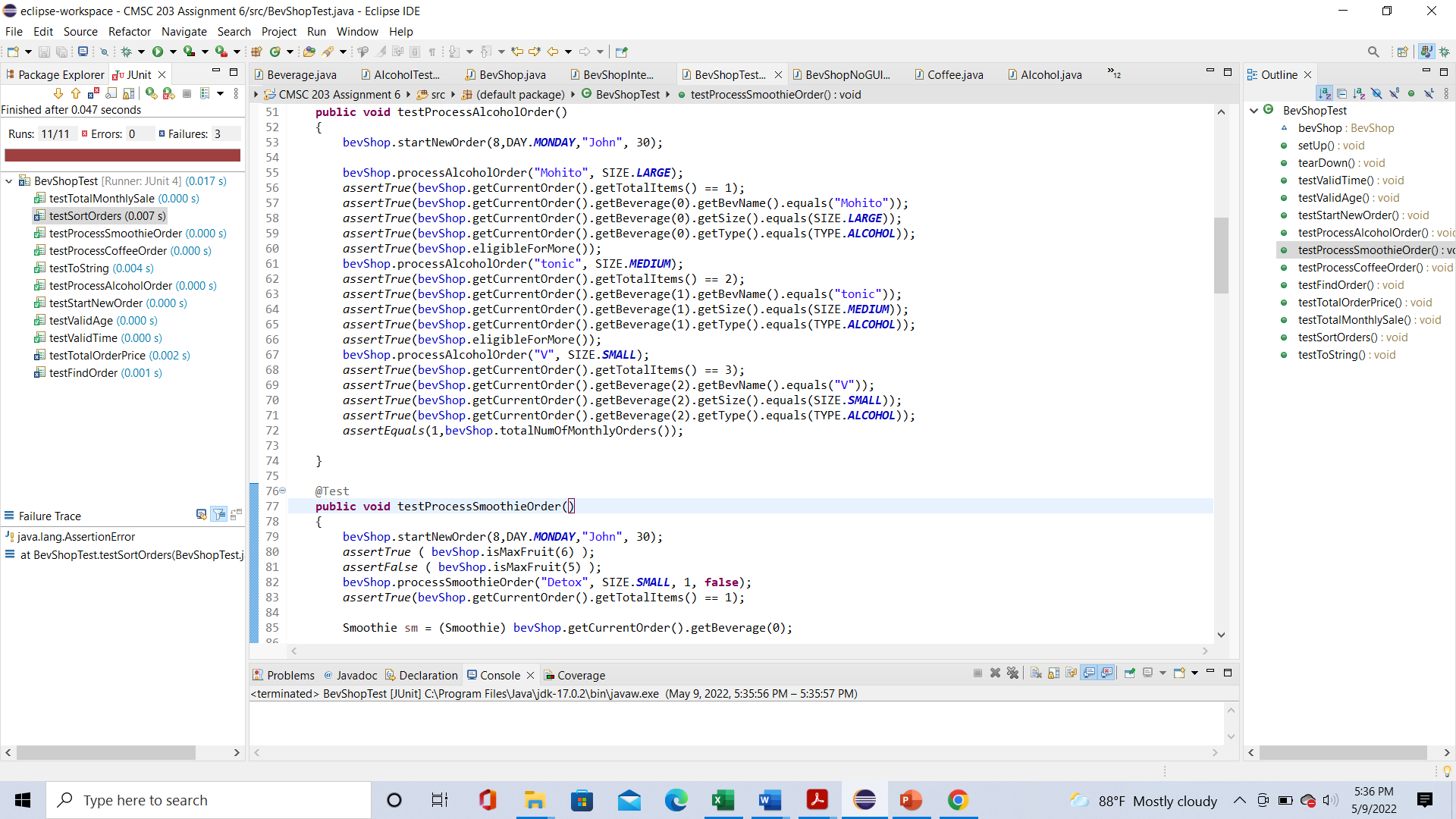
 I have not copied the code from a student or any source.

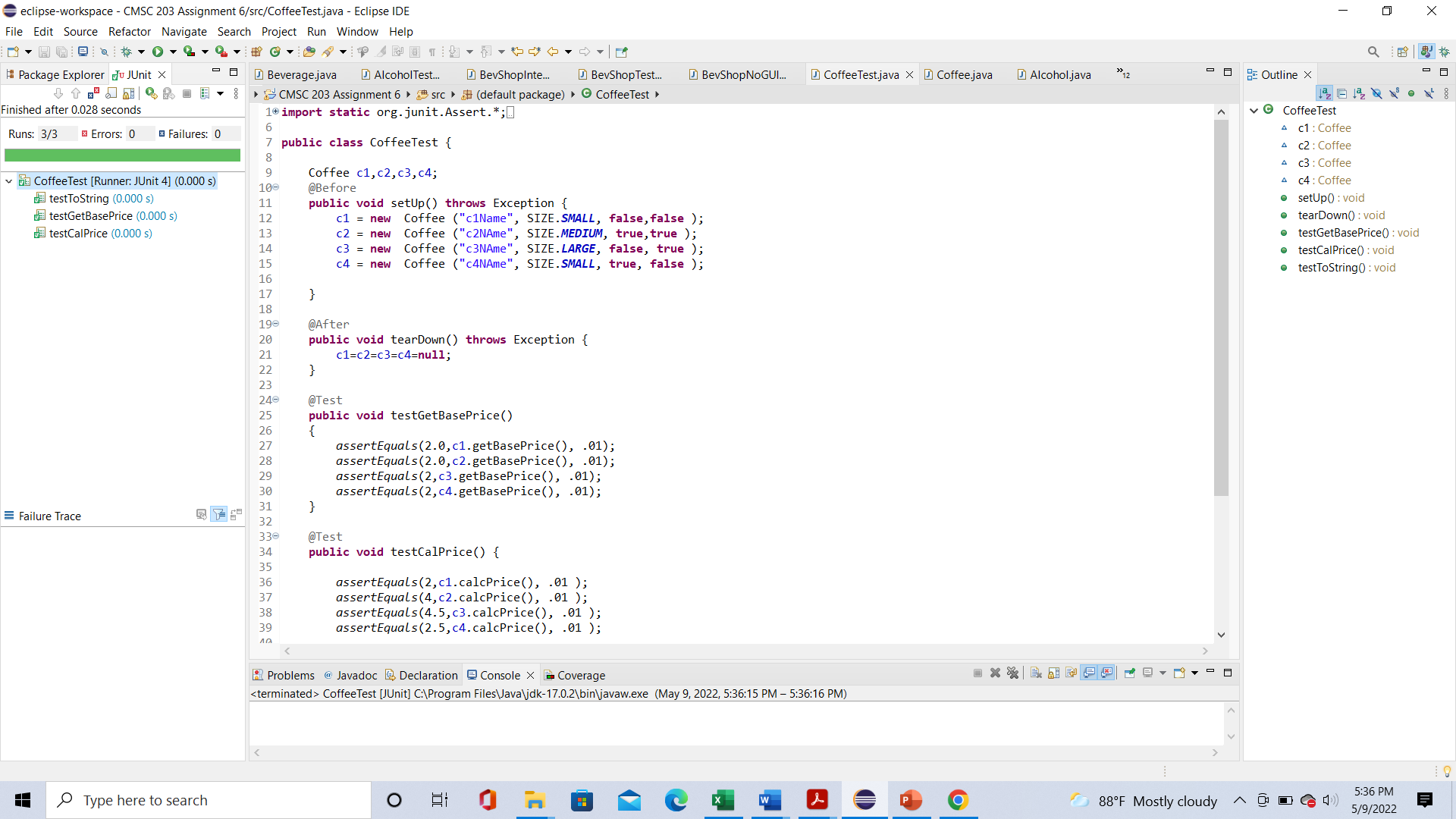
**UML Class Diagram:**

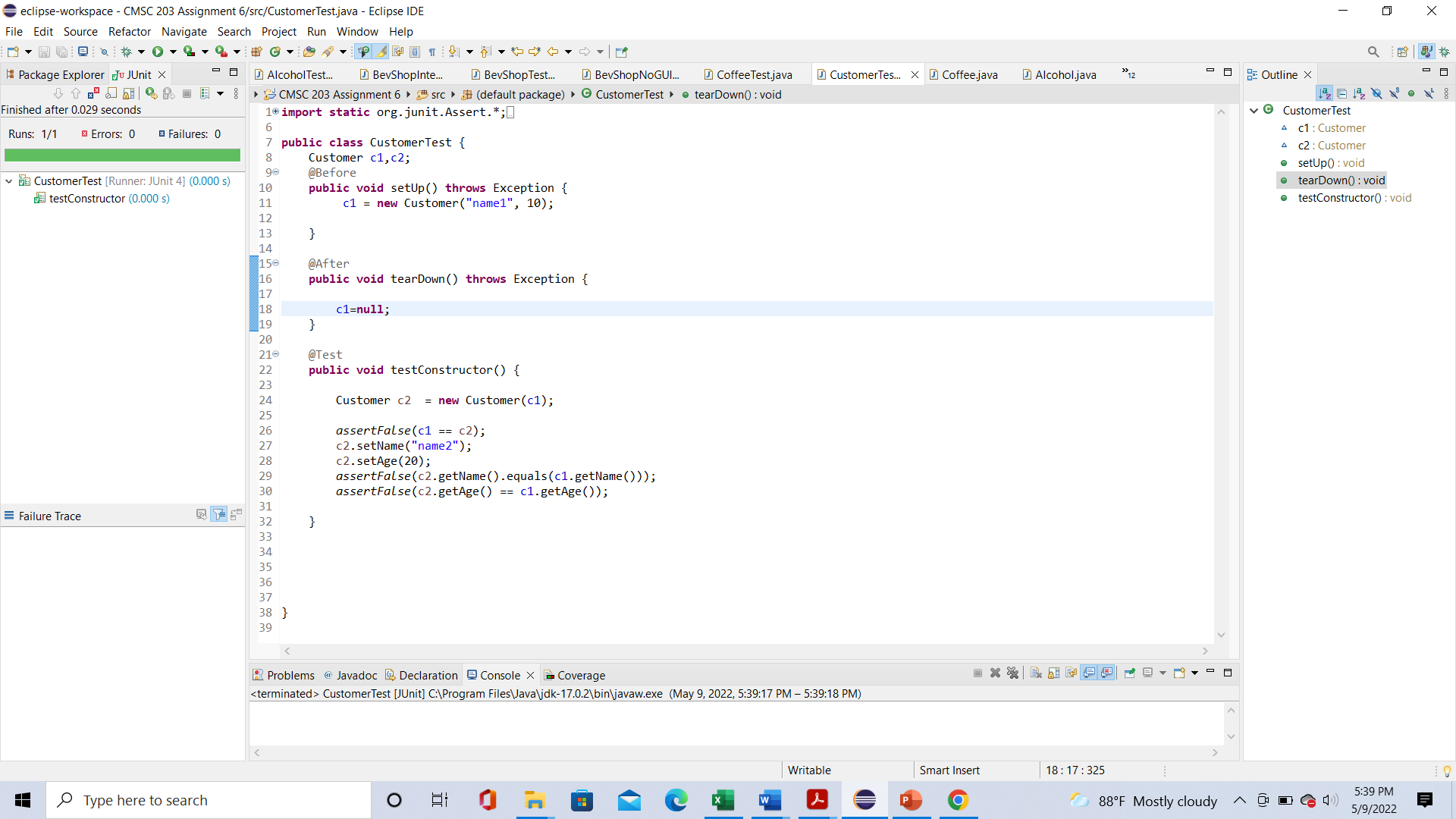


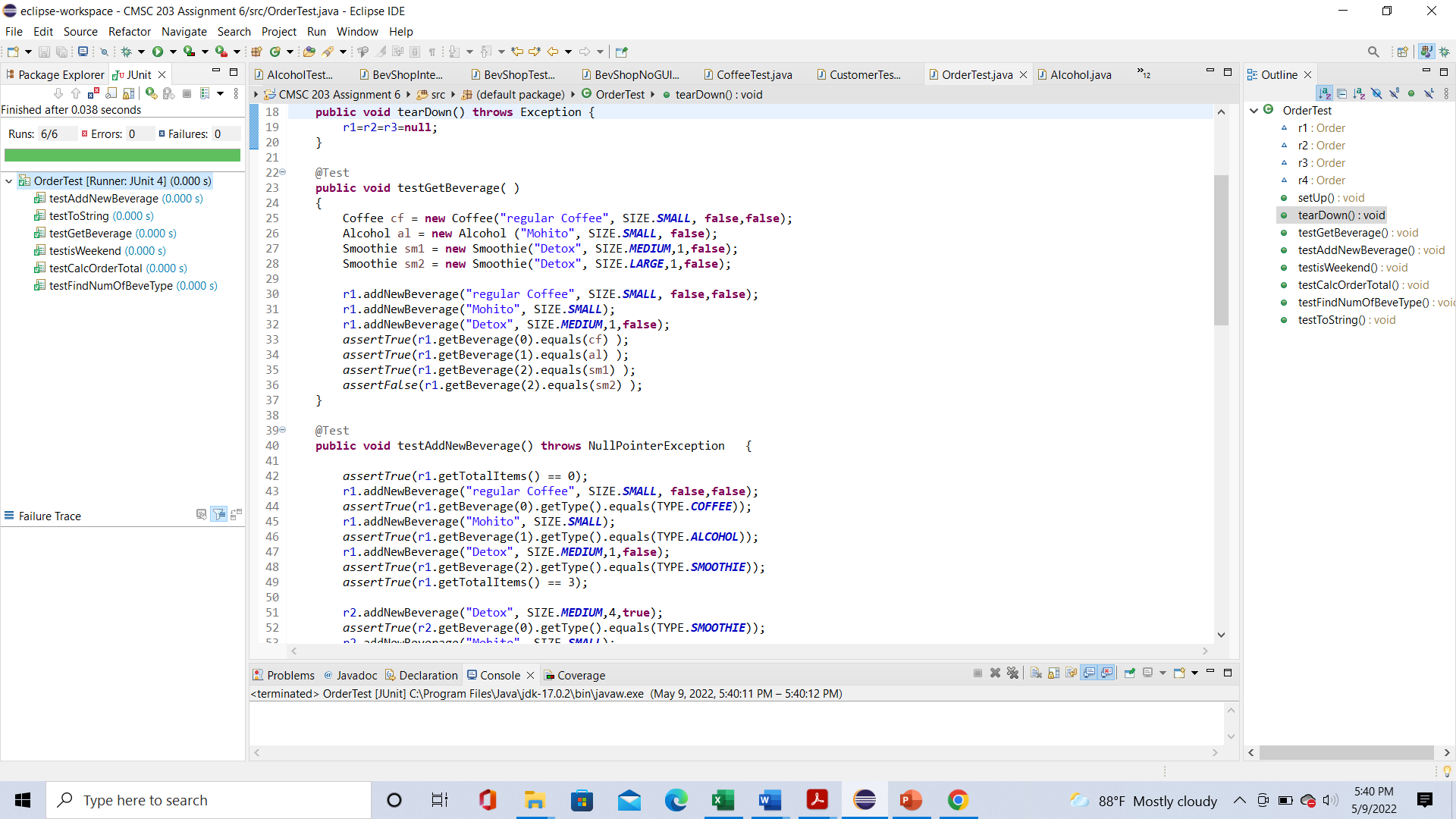
**J\_Unit Tests:**

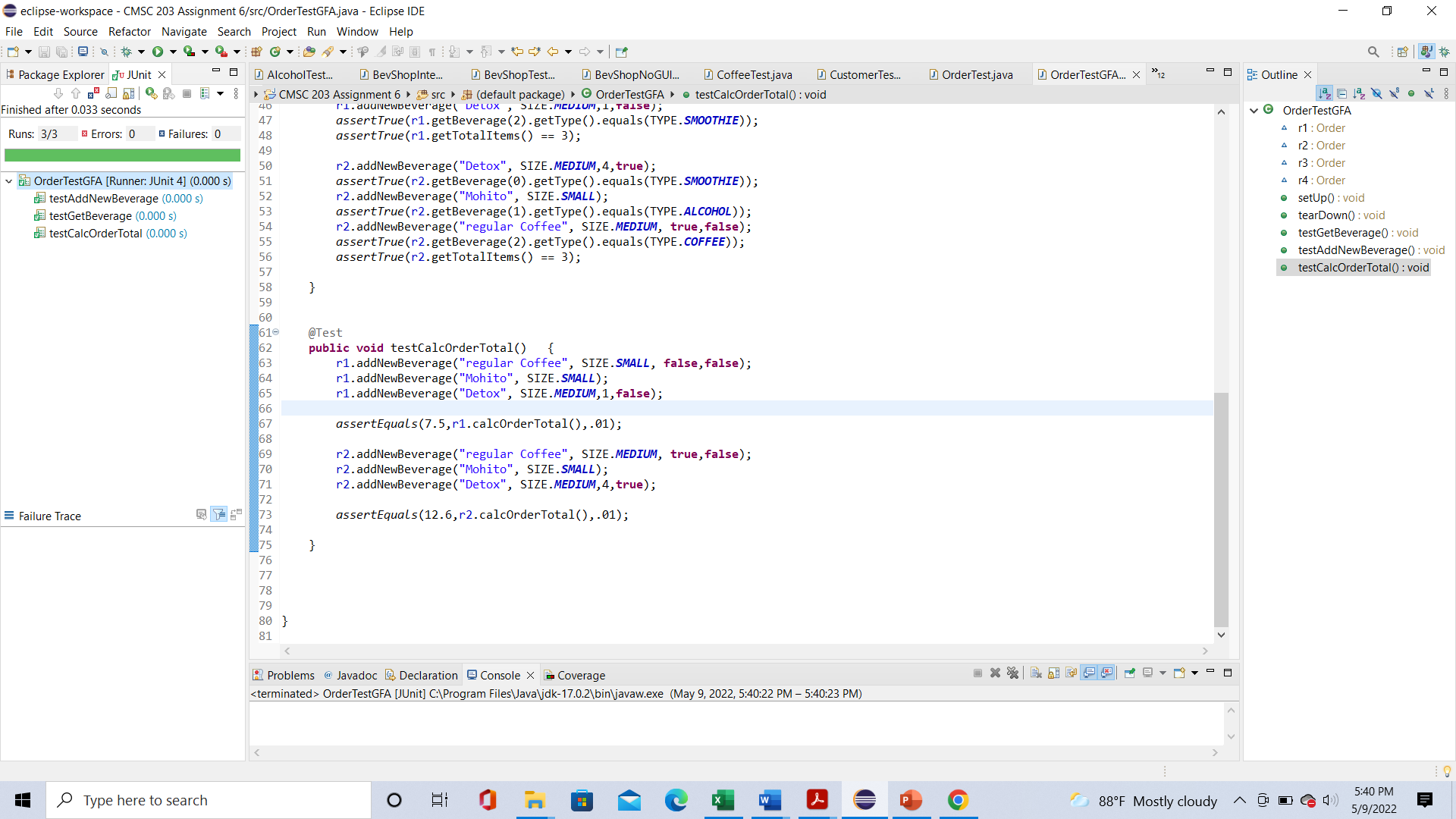
****

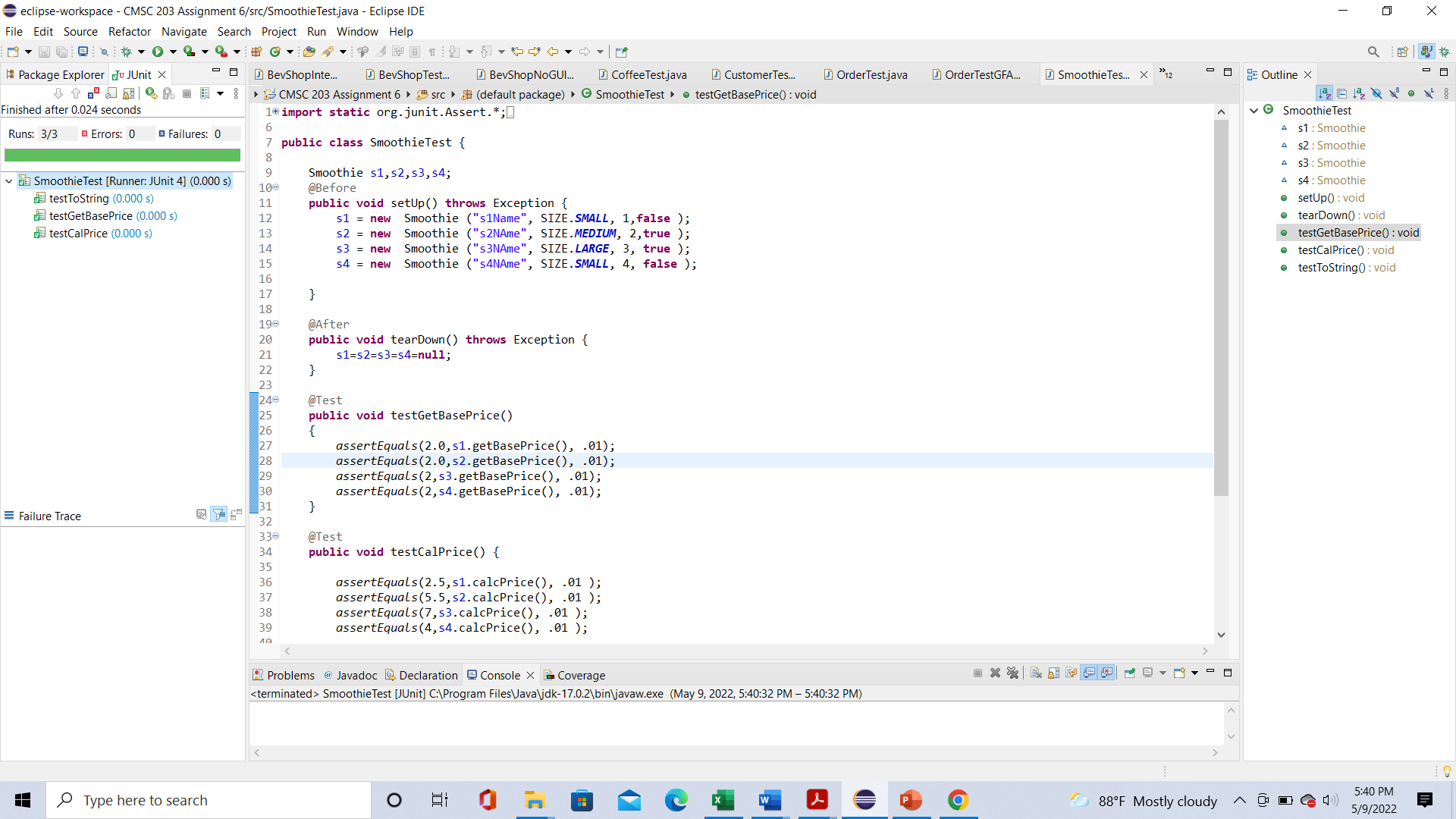
****

****

****

****

****

****

**GUI Run:**

**Graphical user interface, text, application, email

Description automatically generated**

**Git Hub Submission:**

**Lessons Learned** <Provide answers to the questions listed above>**:**

**Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.**

This was probably the toughest assignment. I started by copying and pasting the tests and interfaces from the folder into the src folder of the assignment, making the enum lists, and establishing the java files. Then, I took all the methods from the interfaces and test files and put them into the java files. After that, I created a few getters to make it easier to get some of the program variables. The Boolean methods were simple to set up, but I found that the constructors of the subclasses were not working well. I put a superclass caller and current instances of the variables in the constructors. Also, I didn’t understand what to do for the Comparable Interface.

After some revision form several tutors. I found that I made a few mistakes when it came also realized that the instance variables all needed to be private. I also restated the constructors so that I didn’t need to create second constructors for the Beverage subclasses. This made the Beverage subclass tests pass. The Customer class was simple to set up. However, the Order and BevShop classes were much more difficult. Other than the add and process order methods, the methods were very difficult to set up. I tried to make the methods the best I could, but I had a few problems with getting variables and calling the orderList.

After another look at the instructions and revision at several notes, I realized that I made several mistakes. For Order, I had no variable for the num, and I should try to make sure than there are setters that match the getters I also realized that numOfFruits needs it’s own variable and I needed another order list. I fixed several of these errors and changed the code for a few of my other methods, but eventually, most of my tests passes.

In the end, the only test methods that resulted in errors were testSortOrders(), testTotalOrderPrice(), and testFindOrder(). However, I got the correct GUI and decided that it would be fine these three tests creating errors.

**What have you learned?**

I learned the difference between using arrays and array lists, enumerated lists, implementations, overriding methods, aggregation, inheritance, and polymorphism

**What did you struggle with?**

I mainly struggled with the Beverage, BevShop, and Order classes.

**What would you do differently on your next project?**

I would start my next project sooner and get more help when I could. I think I should also update the java files more often instead of updating them on the same day the project is due.

**What parts of this assignment were you successful with, and what parts (if any) were you not successful with?**

I was successful with the Boolean and getter methods.

I was mainly unsuccessful with the sortOrders method. I think I also didn’t properly understand how to use compareTo() and two of my tests failed because my two of my for loops had breaks in them.

Provide any additional resources/links/videos you used to while working on this assignment/project.

Assignment 6 Check List (include Yes/No or N/A for each item)

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N or N/A** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment6\_Moss.zip | **Y** | **Original java files for classes, subclasses, and enums in this folder** |
|  | * FirstInitialLastName\_Assignment6\_Complete.zip | **Y** | **Html files are in doc folder, java files are in src,** |
|  | **Program compiles** | **Y** | **Program runs** |
|  | **Program runs with desired outputs related to a Test Plan** | **Y** | **Took a lot of tries, but I finally got the same output as the GUI Test** |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **N/A** | **Test plan not needed** |
|  | * Screenshots for each Junit Test | **Y** | **Only the testSortOrders(), testTotalOrderPrice(), and testFindOrder() in the bevShopTest returned as errors. The rest succeeded** |
|  | * Screenshots for each Test case listed in the Test Plan | **N/A** | **Not needed** |
|  | * Screenshots of your java file BevShopNoGUITest run | **Y** | **Included screenshot** |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) |  |  |
|  | * UML Diagram | **Y** | **May be a little messy** |
|  | * Lessons Learned | **Y** | **The** |
|  | * Checklist is completed and included in the Documentation |  |  |

Additional resources:

# Pearson Revel online platform/textbook: Revel: Java Control Structures through Objects plus (1st Edition)

GeeksForGeeks

Stackoverflow