



WS - Shop/src/BevShopDriverApp.java - Eclipse IDE

```
20 * Class: CMSC203
12 import java.util.ArrayList;
13
14 public class BevShopDriverApp {
15     public static void main (String[] args) {
16
17         //initialization
18         ArrayList<Beverage> orderList = new ArrayList<Beverage>();
19         int maxAlcohol = 3;
20         int minAlcoholAge = 21;
21         Customer customer1 = new Customer("John", 23);
22
23         Start a new order
24         Total on the Order: 0.0
25         John
26         23
27         True
28         Add alcohol drink
29         Total on the Order: 2.0
30         True
31         Add second alcohol drink
32         Total on the Order: 6.0
33         2
34         Add third alcohol drink
35         Total on the Order: 8.0
36         3
37         Maximum alcohol drink for this order
38         Add a COFFEE to order
39         3
40
41         Total on the Order: 11.0
42         Total on the Order: 11.0
43         Start a new order
44         Total on the Order: 0.0
45         Add a SMOOTHIE to order
46         Total on the Order: 6.5
47         Total on the Order: 9.5
48         0
49         Age not appropriate for alcohol drink!!
50         Total on the Order: 9.5
51         Maximum number of fruits
52
53         Total on the Order: 16.0
54         Total amount for all the Order: 27.0
55
56 }
```

WS - Shop/src/OrderTestGFA.java - Eclipse IDE

```
49
50     assertTrue(orderOne.getTotalItems() == 0);
51     orderOne.addNewBeverage("regular Coffee", Size.SMALL, false);
52     assertEquals(orderOne.getBeverage(0).getType(), Type.REGULAR);
53     orderOne.addNewBeverage("Mohito", Size.SMALL);
54     assertEquals(orderOne.getBeverage(1).getType(), Type.REGULAR);
55     orderOne.addNewBeverage("Detox", Size.MEDIUM, 1, false);
56     assertEquals(orderOne.getBeverage(2).getType(), Type.DETOX);
57     assertEquals(orderOne.getTotalItems() == 3);
58
59     orderTwo.addNewBeverage("Detox", Size.MEDIUM, 4, true);
60     assertEquals(orderTwo.getBeverage(0).getType(), Type.DETOX);
61     orderTwo.addNewBeverage("Mohito", Size.SMALL);
62     assertEquals(orderTwo.getBeverage(1).getType(), Type.REGULAR);
63     orderTwo.addNewBeverage("regular Coffee", Size.MEDIUM, true);
64     assertEquals(orderTwo.getBeverage(2).getType(), Type.REGULAR);
65     assertEquals(orderTwo.getTotalItems() == 3);
66
67 }
68
69 @Test
70 public void testCalcOrderTotal() {
71     orderOne.addNewBeverage("regular Coffee", Size.SMALL, false);
72     orderOne.addNewBeverage("Mohito", Size.SMALL);
73     orderOne.addNewBeverage("Detox", Size.MEDIUM, 1, false);
74
75     assertEquals(7.5, orderOne.calcOrderTotal(), .01);
76
77     orderTwo.addNewBeverage("regular Coffee", Size.MEDIUM, true);
78     orderTwo.addNewBeverage("Mohito", Size.SMALL);
79     orderTwo.addNewBeverage("Detox", Size.MEDIUM, 4, true);
80
81     assertEquals(12.6, orderTwo.calcOrderTotal(), .01);
82
83 }
84
85 }
```

WS - Shop/src/TypeTestStudent.java - Eclipse IDE

```
1+ import static org.junit.jupiter.api.Assertions.*;
2
3 class TypeTestStudent {
4
5     Type typeA;
6     Type typeS;
7     Type typeC;
8
9     @BeforeEach
10    void setUp() throws Exception {
11         typeA = Type.ALCOHOL;
12         typeS = Type.SMOOTHIE;
13         typeC = Type.COFFEE;
14     }
15
16     @AfterEach
17    void tearDown() throws Exception {
18         typeA = typeS = typeC = null;
19     }
20
21     @Test
22    void test() {
23         assertTrue(typeA.toString().compareTo("ALCOHOL") == 0);
24         assertTrue(typeS.toString().compareTo("SMOOTHIE") == 0);
25         assertTrue(typeC.toString().compareTo("COFFEE") == 0);
26     }
27 }
28 }
```

Finished after 0.062 seconds

Runs: 1 / Errors: 0 / Failures: 0

TypeTestStudent [Runner: JUnit] test() (0.000 s)

Failure Trace

Console X <terminated> TypeTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.jdk/Contents/

WS - Shop/src/SmoothieTestStudent.java - Eclipse IDE

```
1+ import static org.junit.jupiter.api.Assertions.*;
2
3 class SmoothieTestStudent {
4
5     Smoothie smoothie1;
6     Smoothie smoothie2;
7
8     @BeforeEach
9     void setUp() throws Exception {
10         smoothie1 = new Smoothie("Michael", Size.MEDIUM, 0, false);
11         smoothie2 = new Smoothie("Bob", Size.MEDIUM, 4, true);
12     }
13
14     @AfterEach
15     void tearDown() throws Exception {
16         smoothie1 = smoothie2 = null;
17     }
18
19     @Test
20     void testCalcPrice() {
21         assertEquals(3.0, smoothie1.calcPrice(), .01);
22         assertEquals(6.5, smoothie2.calcPrice(), .01);
23     }
24
25     @Test
26     void testToString() {
27         assertEquals("Michael, MEDIUM $3.0", smoothie1.toString());
28         assertEquals("Bob, MEDIUM Protein, 4 Number of Fruits, $6", smoothie2.toString());
29     }
30
31     @Test
32     void testGetNumOfFruits() {
33         assertEquals(0, smoothie1.getNumOfFruits());
34         assertEquals(4, smoothie2.getNumOfFruits());
35     }
36
37     @Test
38     void testGetAddProtein() {
39         smoothie1.setAddProtein(true);
40         assertEquals(true, smoothie1.isAddProtein());
41     }
42 }
```

SmoothieTestStudent [Runner: JUnit] testCalcPrice() (0.000 s)

SmoothieTestStudent [Runner: JUnit] testToString() (0.000 s)

SmoothieTestStudent [Runner: JUnit] testGetNumOfFruits() (0.000 s)

SmoothieTestStudent [Runner: JUnit] testGetAddProtein() (0.000 s)

Failure Trace

Console X <terminated> SmoothieTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.jdk/Contents/

WS - Shop/src/SizeTestStudent.java - Eclipse IDE

```
1 import static org.junit.jupiter.api.Assertions.*;
2
3 class SizeTestStudent {
4     Size sizeS;
5     Size sizeM;
6     Size sizeL;
7
8     @BeforeEach
9     void setUp() throws Exception {
10         sizeS = Size.SMALL;
11         sizeM = Size.MEDIUM;
12         sizeL = Size.LARGE;
13     }
14
15     @AfterEach
16     void tearDown() throws Exception {
17         sizeS = sizeM = sizeL = null;
18     }
19
20     @Test
21     void test() {
22         assertEquals(sizeS.toString().compareTo("SMALL"), 0);
23         assertEquals(sizeM.toString().compareTo("MEDIUM"), 0);
24         assertEquals(sizeL.toString().compareTo("LARGE"), 0);
25     }
26 }
```

Finished after 0.069 seconds

Runs: 1 Errors: 0 Failures: 0

SizeTestStudent [Runner: JUnit] test() (0.001 s)

Failure Trace

Console X <terminated> SizeTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/H

WS - Shop/src/OrderTestStudent.java - Eclipse IDE

```
1 import static org.junit.Assert.assertEquals;
2 import static org.junit.Assert.assertFalse;
3 import static org.junit.Assert.assertTrue;
4 import static org.junit.jupiter.api.Assertions.*;
5
6 import org.junit.jupiter.api.AfterEach;
7 import org.junit.jupiter.api.BeforeEach;
8 import org.junit.jupiter.api.Test;
9
10 class OrderTestStudent {
11     Order order1, order2;
12
13     @BeforeEach
14     void setUp() throws Exception {
15         order1 = new Order(10, Day.MONDAY, new Customer("Michael"));
16         order2 = new Order(18, Day.SATURDAY, new Customer("Bob"));
17     }
18
19     @AfterEach
20     void tearDown() throws Exception {
21         order1 = order2 = null;
22     }
23
24
25     @Test
26     void testGetOrderTime() {
27         assertEquals(10, order1.getOrderTime());
28         assertEquals(18, order2.getOrderTime());
29     }
30
31     @Test
32     void testGetOrderDay() {
33         assertEquals(Day.MONDAY, order1.getOrderDay());
34         assertEquals(Day.SATURDAY, order2.getOrderDay());
35     }
36
37     @Test
38     void testGetDay() {
39 }
```

Finished after 0.058 seconds

Runs: 8 Errors: 0 Failures: 0

CustomerTest... DayTestStude... OrderTestStu... OrderTestStudent [Runner: JUnit] testGetOrderTime() (0.000 s) testGetOrderDay() (0.000 s) testGetDay() (0.000 s)

Failure Trace

Console X <terminated> OrderTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/H

WS - Shop/src/DayTestStudent.java - Eclipse IDE

```
1 import static org.junit.jupiter.api.Assertions.*;
2
3 import org.junit.jupiter.api.AfterEach;
4 import org.junit.jupiter.api.BeforeEach;
5 import org.junit.jupiter.api.Test;
6
7 class DayTestStudent {
8     Day monday;
9     Day sunday;
10
11    @BeforeEach
12    void setUp() throws Exception {
13        monday = Day.MONDAY;
14        sunday = Day.SUNDAY;
15    }
16
17    @AfterEach
18    void tearDown() throws Exception {
19    }
20
21    @Test
22    void test() {
23        assertTrue(monday.toString().compareTo("MONDAY") == 0);
24        assertTrue(sunday.toString().compareTo("SUNDAY") == 0);
25    }
26
27 }
28
29
```

Failure Trace

Console X <terminated> DayTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.jdk/Contents/H

WS - Shop/src/CustomerTestStudent.java - Eclipse IDE

```
1 Customer customer1;
2 Customer customer2;
3
4 @BeforeEach
5 void setUp() throws Exception {
6     customer1 = new Customer("Michael", 24);
7     customer2 = new Customer("Bob", 18);
8 }
9
10 @AfterEach
11 void tearDown() throws Exception {
12     customer1 = customer2 = null;
13 }
14
15 @Test
16 void testGetAge() {
17     assertEquals(24, customer1.getAge());
18     assertEquals(18, customer2.getAge());
19 }
20
21 @Test
22 void testGetName() {
23     assertEquals("Michael", customer1.getName());
24     assertEquals("Bob", customer2.getName());
25 }
26
27 @Test
28 void testToString() {
29     assertEquals("Michael, 24", customer1.toString());
30     assertEquals("Bob, 18", customer2.toString());
31 }
```

Failure Trace

Console X <terminated> CustomerTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.jdk/Content

WS - Shop/src/CoffeeTestStudent.java - Eclipse IDE

```
1 import static org.junit.jupiter.api.Assertions.*;
2
3 class CoffeeTestStudent {
4
5     Coffee coffee1;
6     Coffee coffee2;
7
8     @BeforeEach
9     void setUp() throws Exception {
10         coffee1 = new Coffee("Michael", Size.MEDIUM, false, false);
11         coffee2 = new Coffee("Bob", Size.MEDIUM, true, true);
12     }
13
14     @AfterEach
15     void tearDown() throws Exception {
16         coffee1 = coffee2 = null;
17     }
18
19     @Test
20     void testCalcPrice() {
21         assertEquals(3.0, coffee1.calcPrice(), .01);
22         assertEquals(4.0, coffee2.calcPrice(), .01);
23     }
24
25     @Test
26     void testToString() {
27         assertEquals("Michael, MEDIUM $3.0", coffee1.toString());
28         assertEquals("Bob, MEDIUM Extra Shot, Extra Syrup, $4.0", coffee2.toString());
29     }
30
31     @Test
32     void testGetExtraShot() {
33         assertTrue(coffee2.getExtraShot());
34     }
35
36     @Test
37     void testGetExtraSyrup() {
38         assertFalse(coffee1.getExtraSyrup());
39     }
40
41     @Test
42     void testGetExtraSyrup() {
43         assertFalse(coffee1.getExtraSyrup());
44     }
45 }
```

Finished after 0.066 seconds

Runs: 5 Errors: 0 Failures: 0

CoffeeTestStudent [Runner: JUnit] 0.001s

testGetString() (0.001s)

testCalcPrice() (0.000 s)

testEqualsCoffee() (0.000 s)

testGetExtraShot() (0.000 s)

Failure Trace

Console X <terminated> CoffeeTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents

WS - Shop/src/BevShopTestStudent.java - Eclipse IDE

```
1 import static org.junit.jupiter.api.Assertions.*;
2
3 class BevShopTestStudent {
4
5     BevShop bev1;
6     BevShop bev2;
7
8     @BeforeEach
9     void setUp() throws Exception {
10         bev1 = new BevShop();
11         bev2 = new BevShop();
12     }
13
14     @AfterEach
15     void tearDown() throws Exception {
16         bev1 = bev2 = null;
17     }
18
19     @Test
20     void testIsValidTime() {
21         assertFalse(bev1.isValidTime());
22         assertTrue(bev1.isValidTime(7));
23     }
24
25     @Test
26     void testGetMaxNumOfFruits() {
27         assertEquals(5, bev1.getMaxNumOfFruits());
28     }
29
30     @Test
31     void testGetMinAgeForAlcohol() {
32         assertEquals(21, bev1.getMinAgeForAlcohol());
33     }
34
35     @Test
36     void testIsMaxFruit() {
37         assertTrue(bev1.isMaxFruit(5));
38         assertFalse(bev1.isMaxFruit(3));
39     }
40
41     @Test
42     void testIsEligibleForMore() {
43         assertFalse(bev1.isEligibleForMore());
44     }
45 }
```

Finished after 0.066 seconds

Runs: 8 Errors: 0 Failures: 0

BevShopTestStudent [Runner: JUnit] 0.001s

testIsValidTime() (0.000 s)

testGetMaxFruit() (0.000 s)

testIsValidTime() (0.000 s)

testGetNumOfAlcoholDrink() (0.000 s)

testGetMinAgeForAlcohol() (0.000 s)

testIsValidAge() (0.000 s)

testGetMaxNumOfFruits() (0.000 s)

testGetMaxOrderForAlcohol() (0.000 s)

Failure Trace

Console X <terminated> BevShopTestStudent [JUnit] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents

WS - Shop/src/AlcoholTestStudent.java - Eclipse IDE

```
import static org.junit.jupiter.api.Assertions.*;  
  
class AlcoholTestStudent {  
    Alcohol alcohol;  
    Alcohol alcohol2;  
  
    @BeforeEach  
    void setUp() throws Exception {  
        alcohol = new Alcohol("Michael", Size.MEDIUM, true);  
        alcohol2 = new Alcohol("Bob", Size.MEDIUM, false);  
    }  
  
    @AfterEach  
    void tearDown() throws Exception {  
        alcohol = alcohol2 = null;  
    }  
  
    @Test  
    void testCalcPrice() {  
        assertEquals(3.6, alcohol.calcPrice(), .01);  
        assertEquals(3.0, alcohol2.calcPrice(), .01);  
    }  
  
    @Test  
    void testToString() {  
        assertEquals("Michael, MEDIUM Weekend, $3.6", alcohol.toString());  
        assertEquals("Bob, MEDIUM Weekday, $3.0", alcohol2.toString());  
    }  
  
    @Test  
    void testEqualsAlcohol() {  
        assertFalse(alcohol.equals(alcohol2));  
    }  
  
    @Test  
    void testIsWeekend() {  
        assertTrue(alcohol.isWeekend());  
    }  
}
```

Finished after 0.054 seconds  
Runs: 4 Errors: 0 Failures: 0

AlcoholTestStudent [Runner: JUnit X]

Failure Trace

Console

```
<terminated> AlcoholTestStudent [JUnit] /Library/Java/JavaVirtualMachines/dk-18.0.1.1.jdk/Content
```

bushmn/CMSC203\_Assignment6 (Private)

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file Add file <> Code About

No description, website, or topics provided.

0 stars 1 watching 0 forks

Add files via upload e8ab928 15 seconds ago 1 commit

File	Action	Time	Commit
Alcohol.java	Add files via upload	15 seconds ago	
AlcoholTestStudent.java	Add files via upload	15 seconds ago	
BevShop.java	Add files via upload	15 seconds ago	
BevShopDriverApp.java	Add files via upload	15 seconds ago	
BevShopInterface.java	Add files via upload	15 seconds ago	
BevShopTestStudent.java	Add files via upload	15 seconds ago	
Beverage.java	Add files via upload	15 seconds ago	
Coffee.java	Add files via upload	15 seconds ago	
CoffeeTestStudent.java	Add files via upload	15 seconds ago	
Customer.java	Add files via upload	15 seconds ago	
CustomerTestStudent.java	Add files via upload	15 seconds ago	
Day.java	Add files via upload	15 seconds ago	
DayTestStudent.java	Add files via upload	15 seconds ago	
Order.java	Add files via upload	15 seconds ago	
OrderInterface.java	Add files via upload	15 seconds ago	
OrderTestSE.java	Add files via upload	15 seconds ago	

Releases No releases published Create a new release

Packages No packages published Publish your first package

[https://github.com/bushmn/CMSC203\\_Assignment6](https://github.com/bushmn/CMSC203_Assignment6)

### Overall Thoughts:

This project seemed to pretty smoothly. I did run into a few hiccups with calculating the order price. The alcohol calcPrice() was always off by 60 cents since it didn't recognize that it was the weekend. I think this was an issue with the order method's since the addNewBeverage method should have include boolean weekend, not just name and size. I did a quick fix by adding an if statement to check if the order day matches either Saturday or Sunday and if it does, then add 60 cents to the total.

I learned a lot about passing information from the sub classes and passing it to the Beverage classes. I got comfortable with using enumerated classes. I got much much better at writing junits and how to write different assert equals and asserttrue/false tests for each method. The toString method's were somewhat annoying to create Junits for. It's usually that tiny space or comma that screws up the results. I think the hardest parts were making sure the array list's worked properly with the order numbers, searching through the array list as well as making sure the beverage methods work with the array lists.