Dreams don't work unless you do. **John C. Maxwell**

CPS 2232 - Data Structures

HW #1. Interfaces in Java.

Due Date: Posted on Canvas

Contents

Note: you can change Suchi and Soup to other Food of your preference

1. Create an Interface Edible that has two methods: abstract method howToEat() and default method howToCook() – both return a String.

TASK 1. Provide a UML diagram of your Interface, provide all source-code in a form that can be copy-pasted (so I can run it, not a picture), provide program(s) output (if any) as a screenshot (an image).

2. Create classes Sushi and Soup that both has at least one unique instance data field and an instance method (choose the data field and its type yourself).

TASK 2. Provide all source-code in a form that can be copy-pasted (so I can run it, not a picture), test both of your classes, provide program(s) output (if any) as a screenshot (an image).

3. Add getters, setters and at least two constructors to both classes.

TASK 3. Provide all source-code in a form that can be copy-pasted (so I can run it, not a picture), test all new methods for both of your classes, provide program(s) output (if any) as a screenshot (an image).

4. Add toString() method to both classes.

TASK 4. Provide all source-code in a form that can be copy-pasted (so I can run it, not a picture), test the new method for both of your classes, provide program(s) output (if any) as a screenshot (an image).

5. Both classes should implement the Edible Interface.

TASK 5. Provide UML diagrams of your interface and both of your classes. Provide all source-code in a form that can be copy-pasted (so I can run it, not a picture), test the new method(s) for both of your classes, provide program(s) output (if any) as a screenshot (an image).

6. Create two instances of Sushi and two instances of Soup and test all available functionality (all their methods). Create an array of size 4 of type Edible and add all of these into the array.

TASK 6. Provide all the source-code in a form that can be copy-pasted (so I can run it, not a picture), print all 4 array elements using the for-each loop formatted as a table, provide program(s) output (if any) as a screenshot (an image).

7. Self – tests

Practice the following self-tests until you have at least 70% of questions answered correctly.
Provide a screenshot of results summary displayed after submission for both tests.
https://liveexample-ppe.pearsoncmg.com/selftest/selftest12e?chapter=9&username=liang12e
https://liveexample-ppe.pearsoncmg.com/selftest/selftest12e?chapter=11&username=liang12e
https://liveexample-ppe.pearsoncmg.com/selftest/selftest12e?chapter=13&username=liang12e

Hint: this test can be retaken many times – open it in another tab to start over.

TASK 7. Provide screenshots of the summary of all 3 self-tests. The Percentage of correct/right answers should be at least 70%.