Assignment 2
Gift Hierarchy

Before attempting this project, be sure you have completed all of the reading assignments, non-graded exercises, discussions, and assignments to date.

Design and implement Java program as follows:

(1) There will be a *Gift* class with following attributes: id (combination of numbers and letters), size (values S, M, or L), and price

(2) There will be two child classes *FruitBasket* and *SweetsBasket* with the following additional attributes:

- **FruitBasket:** number of fruits, include citrus fruits indication (true or false)
- **SweetsBasket:** include nuts indication (true or false)

(3) The Gift class hierarchy must provide the following Functionality:

- On creation, a gift instance must be given all attribute values except price which must be calculated and number of fruits which must be set by system
- All attribute values can be changed after creation except id and the values that are calculated or set by the system
- Price is calculated as follows:
    - There is a flat fee of 19.99 for S gift, 29.99 for M gift, and 39.99 for L gift.
    - Fruit basket gift has additional fee of 5.99 when it has citrus fruits.
- Number of fruits in a basket is as follows: Small size has 6 fruits, M has 9 fruits, and L has 15 fruits
- Each class must have a method to return or display the class's values to the console

(4) Implement OrderSystem class with main method with following functionality:

- Order a gift
- Change a gift
- Display gift

(5) Your classes must be coded with correct encapsulation: private/protected attributes, get methods, and set methods and value validation

(6) There should be appropriate overloading and overriding methods

(7) OrderSystem should take advantage of the inheritance properties (e.g. use Gift variable regardless which gift instance as appropriate)

Style and Documentation:

Make sure your Java program is using the recommended style such as:

- Javadoc comment up front with your name as author, date, and brief purpose of the program
- Comments for variables and blocks of code to describe major functionality
- Meaningful variable names and prompts
- Class names are written in upper CamelCase
- Constants are written in All Capitals
- Use proper spacing and empty lines to make code human readable

Capture execution:
You should capture and label screen captures associated with compiling your code, and running the a passing and failing scenario for each functionality

Sample run 1:

```
    MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 3

No gift has been ordered yet

    MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 9

Thank you for using the program. Goodbye!
```

Sample run 2:

```
   MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 2

No gift has been ordered yet
```

Sample run 3:

```
   MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 1
Do you want Fruit Basket (1) or Sweets Basket (2): 1
What size do you want: S, M, or L: S
Do you want citrus fruits included? true/false: true

    MENU
```

```
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 3

FruitBasket [numFruits=6 haveCitrus=true size=S id=FB3150 price=25.98]

    MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 2
Current gift size is: S What size do you want? S, M, or L:
M
Current basket citrus=true Do you want citrus fruits included? true/false: false

    MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 3

FruitBasket [numFruits=9 haveCitrus=false size=M id=FB3150 price=29.99]

    MENU
1: Order a Gift Basket
2: Change Gift Basket
3: Display Gift
9: Exit program

Enter your selection : 9

Thank you for using the program. Goodbye!
```

Submission requirements

Deliverables include Java program (.java) and a single Word (or PDF) document. The Java and Word/PDF files should be named appropriately for the assignment (as indicated in the SubmissionRequirements document.

The word (or PDF) document should include screen captures showing the successful compiling and running of each of the test scenario. Test scenarios should include all required functionality of the program. Each screen capture should be properly labeled clearly indicated what the screen capture represents.

Submit your files to Assignment 2 submission area no later than the due date listed in your online classroom.

Grading Rubric:

The following grading rubric will be used to determine your grade:

Attribute	**Level (15-20 points)**	**Level (5-15 points)**	**Level 0 (0 - 5 points)**
The Gift hierarchy classes	Correct or almost correct attributes and inheritance structure	Mistakes in implementation	Missing or significantly incorrect implementation
Calculate price	Correct or almost correct code to calculate price using overriding	Mistakes in implementation	Missing or significantly incorrect implementation
Encapsulation	Correct or almost correct code for encapsulation	Mistakes in implementation	Missing or significantly incorrect implementation
Test class	Correct or almost correct code to meet required functionality	Mistakes in implementation	Missing or significantly incorrect implementation
Program documentation and style, screen captures	Correct or almost correct menu, program comments, identifiers, and screen captures	Mistakes or incomplete menu, documentation and/or style, and screen captures	Missing or significantly incorrect menu, documentation and/or style, or screen captures