

Due:

Tuesday, 10-March-2019 by 23:59

Deliverables:

The following Java file should be submitted to Google Classroom by the due date and time specified above. Submissions received after the deadline will be subject to the late policy described in the syllabus.

- Assignment01_{StudentNumber}.java

Specifications:

Overview: You will continue the program this semester to maintain the inventory for a store. Do not forget your headers with @author and @since information.

Requirements: Write a set of classes according to the following specifications:

1. Product
 - a. Attributes
 - i. Name: String
 - ii. Price: double
 - iii. Count: int
 - b. Methods
 - i. Constructor that takes the name, price, and count as parameters
 - ii. getName(): String and setName(name: String)
 - iii. getPrice(): double and setPrice(price: double)
 - iv. getCount(): int
 - v. addToInventory(amount: int): None – increase count by amount
 1. if amount is negative, do nothing
 - vi. purchase(amount: int): double
 1. decrease count by amount and return the total price (amount X price)
 2. if amount is negative or greater than count, do not change count and return 0
 - vii. toString(): String – “Product {name} has {count} remaining”
2. FoodProduct – a child of Product
 - a. Attributes
 - i. Calories: int
 - ii. Gluten: boolean
 - iii. Dairy: boolean
 - iv. Meat: boolean
 - b. Methods
 - i. Constructor that takes the name, price, count, calories, gluten, dairy, and meat as parameters
 - ii. getCalories(): int and setCalories(calories: int)
 - iii. containsGluten(): boolean, containsDairy(): boolean, and containsMeat(): boolean

3. ClothingProduct – a child of Product
 - a. Attributes
 - i. Size: String
 - b. Methods
 - i. Constructor that takes the name, price, count, and size as parameters
 - ii. getSize() and setSize(size: String)
4. Customer
 - a. Attributes
 - i. Name: String
 - b. Methods
 - i. Constructor that takes the name as parameter
 - ii. getName(): String and setName(name: String)
5. ClubCustomer – a child of Customer
 - a. Attributes
 - i. Phone: String
 - b. Methods
 - i. Constructor that takes the name and Phone as parameters
 - ii. getPhone(): String and setPhone(phone: String)
6. Store
 - a. Attributes
 - i. Name: String
 - ii. Website: String
 - iii. Count: int
 - b. Methods
 - i. Constructor that takes the name, website, and number of products as parameters
 1. Creates an empty array of products of size number of products
 - ii. getName(): String and setName(name: String)
 - iii. getWebsite(): String and setWebsite(website: String)
 - iv. getInventorySize(): int – returns the size of the array of products
 - v. getCount(): int – returns the number of products saved
 - vi. addProduct(product: Product)
 1. Adds the passed product to the last open position in the array
 2. If the array is already full, do nothing
 3. Returns none
 - vii. getProduct(index: int): Product
 1. returns the Product at the position passed
 2. if the index passed is negative or greater than the index of the last entered product, return null

Design: Your program does not require a main method. You are only responsible for creating the six (6) classes described above. An example of how your program should operate is given below:

```
public class Assignment01_123456789 {
    public static void main(String[] args) {
        Store s = new Store("Migros", "www.migros.com.tr", 3);

        Customer c = new Customer("CSE 102");
        System.out.println(c.getName());

        ClubCustomer cc = new ClubCustomer("Club CSE 102", "05551234567");
        System.out.println(cc.getPhone());

        Product p = new Product("Computer", 1000.00, 20);
        FoodProduct fp = new FoodProduct("Snickers", 2, 100, 250, false, true, false);
        ClothingProduct cp = new ClothingProduct("Shoes", 89, 28, "44");

        for(int i = 0; i < s.getInventorySize(); i++)
            System.out.println(s.getProduct(i));
        s.addProduct(p);
        s.addProduct(fp);

        for(int i = 0; i < s.getCount(); i++)
            System.out.println(s.getProduct(i));
        s.addProduct(cp);

        System.out.println(cp.purchase(2));
        if(fp.containsGluten())
            System.out.println("My wife cannot eat or drink " + fp.getName());
        else
            System.out.println("My wife can eat or drink " + fp.getName());
        s.getProduct(0).addToInventory(3);

        for(int i = 0; i < s.getCount(); i++)
            System.out.println(s.getProduct(i));
    }
}
```

```
----jGRASP exec: java Assignment01_123456789
CSE 102
05551234567
null
null
null
Product Computer has 20 remaining
Product Snickers has 100 remaining
178.0
My wife can eat or drink Snickers
Product Computer has 23 remaining
Product Snickers has 100 remaining
Product Shoes has 26 remaining
```

Code: The file you submit will be named Assignment01_{StudentNumber}. You should put all java classes for this assignment inside of this file as discussed in class.

Test: You are responsible for testing your program. It is important to not rely solely on the examples presented in this Assignment description.

Grading:

Google Classroom Submission: If anything is ambiguous, it is your responsibility to ask questions. It is also your responsibility to complete this assignment in a timely manner. Questions regarding this assignment will likely not be answered if received after 17:00 on the due date of the assignment.

Quiz in Lab: There will be a quiz based on this assignment given on 13-March. The result of this quiz will be used to determine your grade on this assignment. **Note:** if you do not attend the lab and take the quiz, your score on this assignment **will be 0**.