National University
of computer and emerging sciences

CL-1004

Object Oriented Programming- Lab

Spring' 2023

BS-SE

Lab Sessional

1 Hour

Note: Plagiarism (either from internet or from someone else) will result in zero marks.

Ketterdam Grocery System:

We are building a grocery store system that includes products, carts, and a checkout system. The system should be able to handle multiple customers, and each customer should be able to add items to their cart, and checkout.

Suppose a customer named Inej wants to buy groceries from our store. She picks up a few products, including apples and bananas. Products are then added to her cart instance. As she adds products to her cart, the system creates Products (apple, mango etc) and adds them to the Cart that belongs to Inej.

Inej decides to buy a large quantity of apples, so she checks the store's discounts and finds a Bulk Discount (a type of discount) that applies to apples. She counts her products that are apple. If the quantity of apples is equal to or more than 10, the system creates a Bulk Discount with a minimum quantity of 10 and a discount percentage of 10%.

Inej can also select products by comparing different products for their attributes. For example, Inej can compare 2 apples to see which one is more red, big. Also, can compare apples with mangos.

When Inej checks out, the Checkout System instance retrieves her Cart instance and calculates the total price of all the products in her cart, including any applicable discounts. The system applies the Bulk Discount object to Inej's apples and calculates the discounted price based on the quantity of apples in her cart.

After paying for her groceries, Inej leaves the store. The system removes her Cart from the Checkout System object's list of carts. The Product inside Inej's cart are also removed, leaving the system in a clean state for the next customer.

Note:

- Identify all the classes, their relations (inheritance, aggregation and composition) and all the necessary attributes and functions.
- Identify cases of recursion and operator overloading (there are hidden somewhere in the case study).
- All the necessary functions (default constructor, parameterized constructor, copy constructor, getter and setters) should be implemented.
- Create a driver function (main function), and call the functions in a way that realizes the above scenario. Do this for 2 customers, having different values.
- Failing to identify all the covered concepts will lead to deduction of marks.

Submission Details:

1. Save all the .h and .cpp files.

2. Make zip file of all the header and source files (Do not create .rar file) with roll no and lab no. e.g. i22-XXXX_OOP_SessionalExam.zip.

3. Submit the zip file on google class room.