



**Faculty of Engineering and Information Technology**  
**Computer Science Department**  
**COMP2311 Project ( Phase 1)**

---

**Project title:** Grocery Store Inventory

**Project objectives:** 1) To apply the object-oriented programming concepts using Java: classes, inheritance, polymorphism, and interfaces.  
2) To investigate different practices for calling methods like method chaining<sup>1</sup>.

**Project Description:**

As your first job fresh out of college, you're hired on as a bagger at the local supermarket. In order to impress the store manager, you decide to write an Inventory program (برنامج الجرد) to help with inventory tracking (تتبع المخزون). Because you have your heart set on moving into the exciting and fast-paced world of grocery software, you hope that your Inventory program will be a prototype for many future grocery software solutions.

Use an `Inventory` class with an `ArrayList` instance variable that will hold a list of objects, which may be either of two kinds:

- (1) Generic items — instances of an `Item` class that includes three instance variables for: type description, quantity of that type, and price of one unit of that type.
- (2) Branded items — instances of a `Brand` class that is derived from the `Item` class and adds an additional instance variable to specify a brand name.

Figure 1 illustrates a suggested UML class diagram:

---

<sup>1</sup> Method Chaining is the practice of calling different methods in a single line instead of calling other methods with the same object reference separately. For more information visits <https://www.geeksforgeeks.org/method-chaining-in-java-with-examples/>

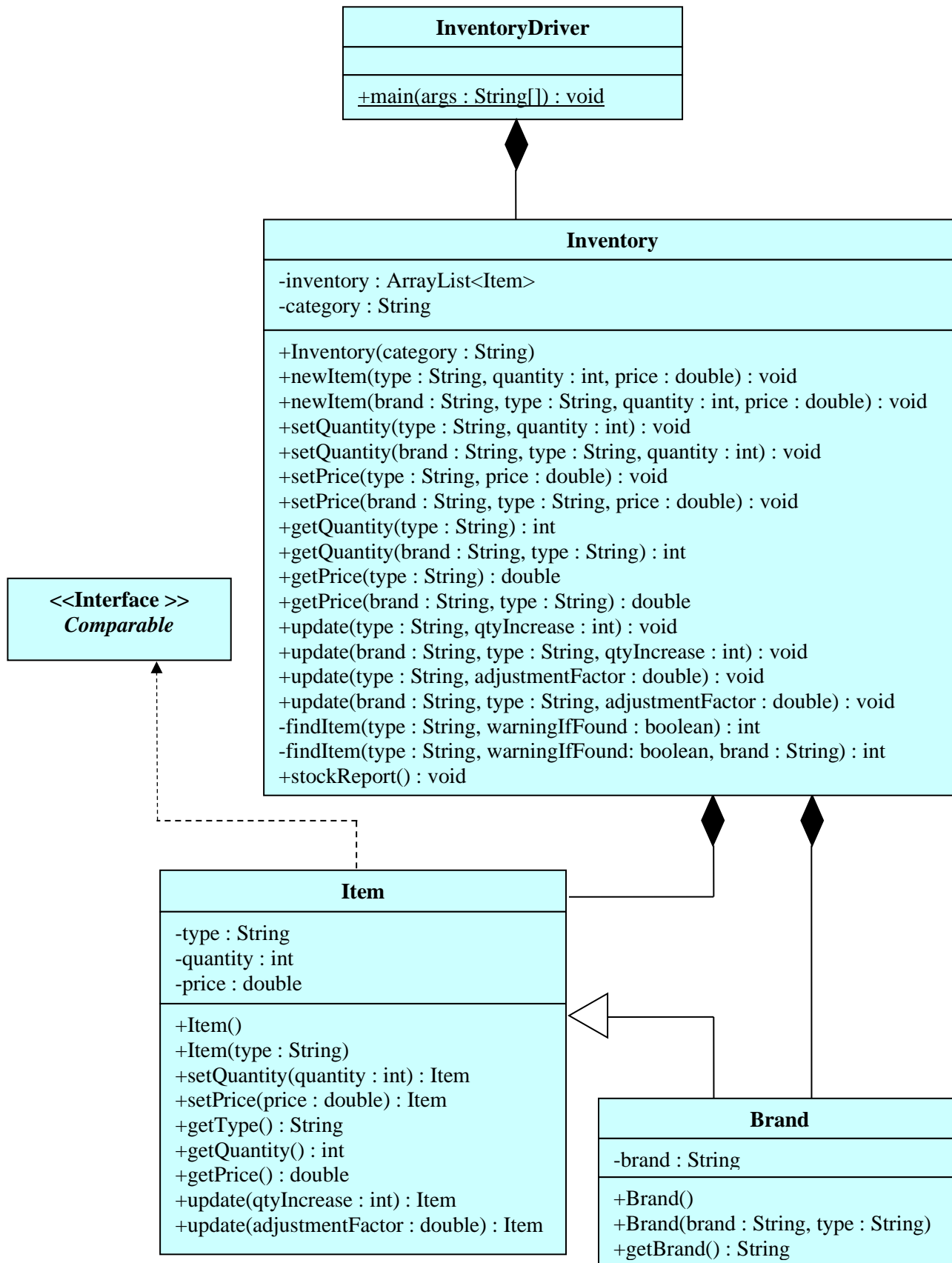


Figure 1: Suggested UML class diagram

In **InventoryDriver**'s **main** method, test your other classes with the following lines of code:

```
Inventory store = new Inventory("groceries");
store.newItem("bread", 15, 9.99);
store.newItem("al-jebrini", "milk", 2, 2.00);
store.newItem("eggs", 3, 1.50);
store.newItem("bread", 2, 1.25); // warning: in stock

store.stockReport();
store.update("al-jebrini", "milk", .25); // raise price 25%
store.update("eggs", -1); // lower quantity by 1
store.update("juice", 3); // warning: not stocked
store.newItem("juneidi", "milk", 4, 1.95);
store.stockReport();
store.setPrice("cola", 10); // warning: not stocked
store.setQuantity("al-jebrini", "milk", 3);
store.setPrice("eggs", 2.00);
System.out.println("milk quantity: " + store.getQuantity("alsafi",
"milk")); // not stocked
System.out.println("milk quantity: " + store.getQuantity("milk"));
// ambiguity
System.out.println("eggs price: " + store.getPrice("eggs"));
System.out.println("milk price: " + store.getPrice("milk")); //
ambiguity
System.out.println((new Item("cola").setPrice(5)).compareTo(new
Item("milk").setPrice(8)));
```

Using the above lines of code, your program should generate the following output.

Output:

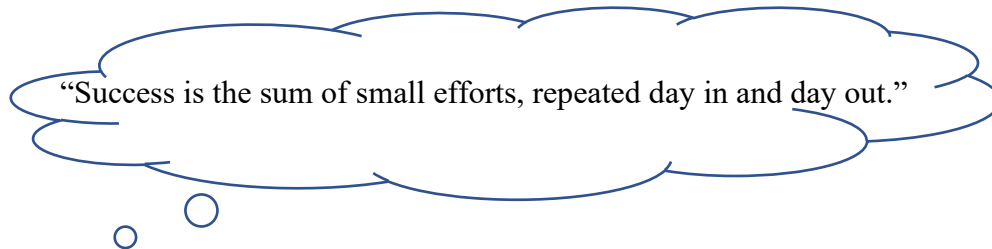
```
bread already exists
bread - in stock: 15, price: $9.99
al-jebrini milk - in stock: 2, price: $2.00
eggs - in stock: 3, price: $1.50
Total value: $158.35

Cannot find juice
bread - in stock: 15, price: $9.99
al-jebrini milk - in stock: 2, price: $2.50
eggs - in stock: 2, price: $1.50
juneidi milk - in stock: 4, price: $1.95
Total value: $165.65

Cannot find cola
Cannot find alsafi milk
milk quantity: 0
Found more than one brand of milk
milk quantity: -1
eggs price: 2.0
Found more than one brand of milk
milk price: NaN
-3
```

**Please note the Followings:**

1. This is an **individual work project**. There will be disciplinary action taken against those who **cheat**.
2. Based on Java formal documentation, your program should be **well commented**.
3. **Submissions:**
  - 3.1 Due date: **Saturday, December 10, 2022, at 12:00 am (Midnight)**.
  - 3.2 Make sure you submit your project through **ITC before the deadline**.
  - 3.3 Your project should be submitted as a **compressed file** in the following **format**: (yourId\_yourName\_Section#), for example, **1020077\_Sandy Ahmad\_Section 4.rar**.



Robert Collier

Best of Luck