

# MILESTONE 1

## Overview

In this milestone, you will develop the UML Class diagram of your software using the information provided in this overview. You will also be coding the beginnings of the *StoreManager*, *Inventory* and *Product* classes. You can find the specific requirements for each class below.

*Keep in mind that most of the tasks you will be required to do are relatively **open-ended**. You must justify in your Change Log any decisions you made that were not obvious. For example, you should give a brief reason why you used an Array instead of an ArrayList for a class field. However, you should not explain why you gave *studentName* a type of *String*. Another example: you do not need to explain why you made a class method *public*, or a class field *private*, because doing this is usually normal. But, if for some reason, you made a class attribute *protected*, you need to explain why.*

## The Product Class

The **Product** class will store information about items being sold by the store. A Product object **must only** have a (1) name, (2) id, and (3) price. It does not have any other fields. The information in these fields should be retrievable but cannot be changed once the Product object is created.

## The Inventory Class

The **Inventory** class will track the state of the inventory of your system. It should keep track of the type and quantity of each Product, as well as provide methods to access and modify this information.

The following functionalities should be available in any given *Inventory* object:

- Get the amount of stock for a given Product ID (Note: it is possible the Product does not exist in the Inventory!).
- Add a specified amount of stock for a given Product to the inventory (Note: new Products can be added!).
- Remove a specified amount of stock for a given Product ID from the inventory (Note: you cannot have negative stock, **and** you cannot delete Products from the Inventory; if a Product's stock reaches 0, leave it.).
- Get information on a Product given a Product ID.
- Initialize the Inventory; set the contents of the Inventory to some default values upon object creation.

## The StoreManager Class

*StoreManager* is the “brain” of the system. It contains all the functionality for managing the *Inventory*, *ShoppingCarts*, and providing information to the *StoreView* class.

*StoreManager* manages a single *Inventory* and it will have a variety of methods – two for now- to interact with this *Inventory* object.

## SYSC 2004 – Course Project

---

A StoreManager object should:

- Create a new *Inventory* object upon object creation (i.e., when a StoreManager object is created).
- Have functionality to check how much stock of a given *Product* is in the *Inventory*.
- Have functionality to process a transaction given an Array of Product information. For example, imagine that the content of a user's shopping cart is as follows: "[[productID1, quantity], [productID2, quantity], [productID3, quantity]]". Given each *productID*, your method should:
  - o Check that the desired quantity exists in the *Inventory* and return the total for all of the Products.
  - o If there is insufficient quantity of any of the products, your method should return some indication of this failure. It could be, for example -1.
  - o Remember to subtract the quantities from the *Inventory* stock if the transaction is successful.

### Questions

1. What is a constructor? When does it get used in Java?
2. Did you need to specify a constructor for the *StoreManager* class? Why or why not?
3. Explain what a default constructor is.
4. What are object references and where are they used in this milestone?
5. Summarize the most important differences between an *ArrayList*, *LinkedList*, and *Array*. Which one did you use in this milestone and why? If you used something else, you must explain why as part of your Change Log.
6. What is encapsulation and how is it relevant to this milestone?

### Milestone 1 Deliverables

1. A PDF named "**storeUML\_M1.pdf**" with the UML class diagram of the current state of the system. For now, only include the **three classes** explained in this milestone.
2. The following classes completed according to Milestone 1 specifications: *Inventory.java*, *StoreManager.java*, *Product.java*. Do not jump ahead!
3. Everything applicable to Milestone 1 from General Submission Requirements. Do not forget the report with the change log and the answers to the questions.