Project 2- Inventory System

Building Blocks

# Objective

Explore Data Abstraction.

# Instructions

1. Create an application called Inventory System that has the following classes:

A *Product* class that minimally stores the following data fields:

* Product ID
* Description
* Price

The following methods should also be provided:

* A constructor that initializes the ID, Description and Unit Price fields
* Methods to set and get the Description
* Methods to set and get the Unit Price
* Method to print all attributes of the Product class

A *Sale Item* class that minimally stores the following data fields:

* Product ID
* Unit Price
* Units

The following methods should also be provided

* A constructor that initializes the Product ID and Units fields
* Methods that set and get the Product ID field
* Methods that set and get the Unit Price field
* Methods that set and get the Units field
* A method that calculates the Total Price for the Sale
* A method to print all attributes of the Sale Item class

A *Sales Receipt* class that minimally stores the following data fields:

* Sale ID
* Customer ID
* Date/Time
* List of Sale Items

The following methods should also be provided:

* A constructor that initializes the Sale ID, Customer ID, and Date/Time fields
* A method to retrieve the Sale ID
* A method to retrieve the Customer ID
* A method to retrieve the Date/Time
* A method to calculate the total for the Sales Receipt
* A method to print all attributes of the Sale Receipt class

1. Read the Sales and Product data from the provided files
2. Print a list of Products sold including the Product ID and Description. In the list include the Unit Price for each Product, the number of units sold and the total revenue for each product.
3. Create the following for the application
4. Draw a UML diagram for this application
5. Identify and document the requirements
6. Create Test Cases that will demonstrate the requirements
7. Implement the previous classes in Java.
8. Write a main program or test class that tests all of the methods created and demonstrates that they are working and meet the requirements.
9. Document your test results

# Submit

Submit the following:

## Week 1 Submission

* List of Requirements
* UML diagram and pseudo-code for the application
* Test Cases to demonstrate requirements

## Final Submission

* Updated list of Requirements
* Updated UML diagram and pseudo-code for the application
* Updated Test Cases
* Text file(s) containing the classes and methods
* Text file containing the Test Results