Java Fundamentals
Section 4: Creating an Inventory Project
Project

Overview

This project will progress with you throughout Sections 4, 5, 6, and 7 of the course. After each section there will be more to add until it builds into a complete Java application to maintain Inventory. For each part, build upon the last part so that both the old and new requirements are met. Include all parts in a package called inventory.

Create an inventory program that can be used for a range of different products (cds, dvds, software, etc.).

Topic(s):

- Data types (Section 4.3)
- Creating classes/objects (Section 4.2)
- Instance variables/fields (Section 4.2)
- Constructors (Section 4.2)
- Methods (getters/accessors, setters/mutators) (Section 4.2)
- Overloading (Section 4.2)
- Main/tester classes (Section 4.2)
- toString() (Section 4.4)

Instructions:

1. For the first part of the project you are required to think about what your inventory system will store.
    a. Think of specific products that lend themselves to be stored in an inventory (for example, products in your home, school, or workplace: they could be from the following categories; office supplies, music CDs, DVD movies, or software). Write a list of at least 6 products that you want to store in your system, this project could be used to store a wide range of products. 1. Sprite 2. Bustelo 3. Rice 4. Sauce 5. tuna-in-can 6. Noodles
    b. For each of the products that you identified, complete the following table:

Attribute	Sample Data
Name of the product **(the value that will identify the product in your system).**	Sprite, Bustelo, Rice, Sauce, tunaCan, Noodles
Price **(this value holds the price that each item will be sold for).**	soda = 2; coffee = 3, rice = 5, sauce = 3, tunaCan = 3, Noodles = 4
Number of units in stock **(this value will store how many of each product item is currently in stock).**	sodaStock = 20, coffeeStock = 20, riceStock= 30 sauceStock = 15, tunaCan = 15, noodleStock = 5
Item number **(used to uniquely identify the product in your system).**	soda#= 1, coffee#= 2, rice#=3, sauce#=4, tunaCan#=5, noodle#=5

This table gives you an understanding of the type of data that you will want to store for the attributes of each product. It's useful to do this so you have a clear understanding of the data that you will be working with!

2. The next step is to think about the correct data types that you will use to store the values in your system. To do this add another column to your table that will identify the correct data type for each value that you have identified.

Attribute	Sample Data	Data Type
Name of the product	Sprite	String
Price	$2	int
Number of units in stock	20	int
Item number	1	int

3. Create a project named **inventory**.

4. Create an object class called **Product**.

5. Add the following private instance fields (variables) by using the data types you identified in task 2:
    a. item number
    b. the name of the product
    c. the number of units in stock
    d. the price of each unit

6. Add a comment above the instance field declarations that states:

```
//Instance field declarations
```

7. Create two constructors:
    a. A default constructor without parameters that will allow the compiler to initialize the fields to their default values. Add a comment above your constructor that explains the purpose of the code.
    b. Overload the default constructor by creating a constructor with parameters for all four of the class' instance fields so that they can be initialized with values from the driver class. The parameters should be named; number, name, qty, price. You should use the *this.instance_field_name* notation to quantify the objects instance field:

```
this.name = name;
```

8. Write getter/accessor and setter/mutator methods for each of the four instance variables. Write getter/accessor and setter/mutator methods for each of the four instance variables. Add comments above them to explain their purpose.

9. Override the toString() method from the object class to show a description of each Product object that includes the instance field values in the following format:

```
Item Number      : 1
Name             : Greatest Hits
Quantity in stock: 25
Price            : 9.99
```

10. Create a Java main class called ProductTester.

11. Create and initialize six Product objects based on the list that you created in task 1.
    a. Two of the Products should be created using the default constructor.
    b. The other four should be created by providing values for the arguments that match the parameters of the constructor.

12. Using the ProductTester class, display the details of each product to the console.

13. Save your project.