

For this project you will design a program that works like a simple cash register. It will be able to accept two different types of items and calculate the total price of any combination of items. The following will provide the necessary information for your program to meet the requirements of this project.

### **Part 1- Interface:**

Write an interface that contains one method that accepts nothing and returns a double. This method will serve to calculate the total price of an individual item in the classes that implement the interface. Choose the method name accordingly.

### **Part 2- Two Classes:**

For this part of the project you will write two classes that implement the interface created in part one. Each class will contain a constructor to initialize class variables and one method that implements the abstract method defined in the interface. As stated, this method will return the price of an item of the class type based on the values passed to the constructor. Write the classes as follows:

- **General Merchandise:** The constructor for the general merchandise class should accept two variables of type double. One variable should represent the price of the item and the other the percent off if the item is on sale. For example, if an item is twenty percent off, 20 should be entered. If the item is not on sale, zero should be passed to the constructor. The constructor parameters should be used to initialize price and percentage instance variables. This class will also contain a method that returns the price of the item, as mentioned previously. To calculate the price of the item, a sales tax of six percent should be considered as well as the percent off, if the item is on sale. Note that the sales tax is always determined after any percent discount off the price.
- **Produce:** The constructor for produce should accept three variables of type double. One variable should represent the price of the item per pound, another should represent the weight in pounds, and the last should represent the percent off (as defined in the general merchandise class). In the method that returns the price of the item, a sales tax of six percent and the percent off the item should be considered.

### **Part 3 – The Driver:**

The driver class in your program should contain code that allows the user to enter different types of items, store them in an array, and then print out the total price of all items in the array. You may design your driver class however you see fit but you must satisfy the following conditions:

- The array that stores the items must be of the type of the interface created in part one. It should be initialized to hold 15 elements.
- The user should be asked what type of item they would like to store in the array as well as all the required information for the constructor of that specific type. Based on this input an object of the input type should be created and its reference stored in the array of items.
- The user should be able to enter as many items as they wish (up to 15) and be able to enter a sentinel to indicate when they have no more items. Use a try/catch structure to print out an error message if the user attempts to enter more than 15 items.
- When the user is done entering items the total price of all items stored in the array should be printed out. Use the Decimal Format class to format output in terms of dollars and cents (i.e. 32.45.)
- The total price of all items may also be calculated using recursion. Write an additional recursive method in the driver class that calculates the sum. The sum calculated from this method should also be printed, be sure to validate your results. Use the Decimal Format class to format output in terms of dollars and cents.

### **Part 4 – The Test:**

Enter the following items into your program. The final total should be \$79.30.

Type	Discount	Price	Price per pound	Weight (lbs)
General	0%	\$2.55	NA	NA
General	25%	\$35.04	NA	NA
General	55%	\$4.75	NA	NA
Produce	0%	NA	\$3.55	3.44
Produce	30%	NA	\$17.65	2.56

