



## Technical Exercise

Implement a solution for the following problem in C# or PHP (v7 or newer) with the intent for the output to be included in a client application maintained by Voyager.

**NOTE:** Please do not use any frameworks (Laravel, Symfony, etc) as these obscure the skills we are assessing.

We are looking for clean, well-factored OO code.

The submission should be a .Net class library or PHP library. You can opt to put a user interface on it if you wish, but we will only be assessing the library portion of your work.

Ideally your submission will be hosted in GitHub or similar cloud hosted version control system but we will accept compressed archives.

You do not need to provide any form of persistence in this application.

Your solution should contain some way of running automated tests to prove that it works.

Consider a grocery market where items have prices per unit but also volume prices. For example, Apples may be \$1.25 each, or 3 for \$3.

Implement a class library or PHP library/package for a point-of-sale scanning system that accepts an arbitrary ordering of products, similar to what would happen at an actual checkout line, then returns the correct total price for an entire shopping cart based on per-unit or volume prices as applicable.

Here are the products listed by code and the prices to use. There is no sales tax.

Product Code	Unit Price	Bulk Price
A	\$1.25	3 for \$3.00
B	\$4.25	
C	\$1.00	\$5 for a six-pack
D	\$0.75	

The interface at the top-level PointOfSaleTerminal service object should look something like this, (written in pseudo-code for illustration purposes)

```
var terminal = new PointOfSaleTerminal()

terminal.SetPricing(...);

terminal.ScanProduct("A");

terminal.ScanProduct("C"); // etc

decimal result = terminal.CalculateTotal();
```

You are free to define the pricing data model as you see fit.

Here are the minimal inputs you should use for your test cases. These test cases must be shown to work in your program:

1. Scan these items in this order: ABCDABA Verify that the total price is \$13.25
2. Scan these items in this order: CCCCCC Verify that the total price is \$6.00
3. Scan these items in this order: ABCD Verify that the total price is \$7.25