



Inbox x



to me ▼

May 2, 2022, 2:19 PM (2 days ago)



Using the Module pattern, implement code for a `groceryStore` Module. Download the attached, `groceryStoreModule.js` file, which contains the starter code, provided for you.

For your solution, you are expected to implement code for the following:

3.1 A private method named, `toString`, in the `Product` class - which Returns the string representation of the Product data in the format, { ProductId: 1001, Name: Apples, UnitPrice: 1.25, QuantityInStock: 236 }.

3.2 A private method named, `getTotalValueInStock`, in the `Product` class - which Returns the total value of the product in stock, calculated by multiplying its `unitPrice` with the quantity in stock.

3.3 A public method of the `groceryStore` Module, named, `upsertProduct` - which adds a new product to the array of products in the store, if it does not already exist, otherwise it updates the existing product data.

3.4 A public method of the `groceryStore` Module, named, `getGrandTotalValueInStock` - which calculates and returns the grand total of the value of all the products in the store. It is the sum of the `TotalValueInStock` for each of the products in the store (Hint: Use `Array.reduce(...)` to implement this method).

3.5 A public method of the `groceryStore` Module, named, `listInventory` - which prints-out to the console the data for each product in the store.

When done implementing the above methods, the following code which is already included in the `groceryStoreModule.js` file, can be used to test your solution; comparing/verifying its correctness based on the expected output, as shown below:

```
groceryStore.upsertProduct(1001, "Apples", 1.25, 236);
groceryStore.upsertProduct(1002, "Bananas", 0.99, 124);
groceryStore.upsertProduct(1003, "Pears", 1.05, 97);
```

```
groceryStore.listInventory();
```

```
// Expected output:
```

```
{ ProductId: 1001, Name: Apples, UnitPrice: 1.25, QuantityInStock: 236 }  
{ ProductId: 1002, Name: Bananas, UnitPrice: 0.99, QuantityInStock: 124 }  
{ ProductId: 1003, Name: Pears, UnitPrice: 1.05, QuantityInStock: 97 }
```

```
groceryStore.upsertProduct(1003, "Pears - organic", 1.55, 107);
groceryStore.listInventory();
```

```
// Expected output:
```

```
{ ProductId: 1001, Name: Apples, UnitPrice: 1.25, QuantityInStock: 236 }
{ ProductId: 1002, Name: Bananas, UnitPrice: 0.99, QuantityInStock: 124 }
{ ProductId: 1003, Name: Pears - organic, UnitPrice: 1.55, QuantityInStock: 107 }
```

```
console.log(`Total Store value: ${groceryStore.getGrandTotalValueInStock()}`);
```



➡ Forward