

Sonatafy - Sales Taxes

1.0.0

Author: Apolo Orihuela

Date Created: 07/08/2021

Document Version: 1.0

Contents

- Section 1 - Project Description
 - 1.1 Project
 - 1.2 Description
 - 1.3 Change History
- Section 2 - Overview
 - 2.1 Purpose
 - 2.2 Scope
 - 2.3 Requirements
 - 2.4 System Requirements
 - 2.5 Debug Mode Notes
- Section 3 - Deployment
 - 3.1 File System Structure
 - 3.2 Console Application Deployment
 - 3.3 Console Application Running Examples

Copyright

© Sonatafy 2021. All rights reserved.

This document and its content remain the property of Sonatafy in no way may any part of it be reproduced, transmitted or stored in any retrieval system without prior written consent, unless expressly permitted in any written contract signed by the recipient and Sonatafy. No part may be disclosed to or used by any third party unless expressly permitted by a relevant written agreement. If no such agreement has been signed by the recipient Sonatafy with You the document and its content shall remain confidential.

Section 1 - Project Description

1.1 Project

Create a Windows Console Application using .Net C#

1.2 Description

There are a variety of items for sale at a store. When a customer purchases items, they receive a receipt. The receipt lists all of the items purchased, the sales price of each item (with taxes included), the total sales taxes for all items, and the total sales price.

Basic sales tax applies to all items at a rate of 10% of the item's list price, with the exception of books, food, and medical products, which are exempt from basic sales tax. An import duty (import tax) applies to all imported items at a rate of 5% of the shelf price, with no exceptions.

Write an application that takes input for shopping baskets and returns receipts in the format shown below, calculating all taxes and totals correctly. When calculating the sales tax, round the value up to the nearest 5 cents. For example, if a taxable item costs \$5.60, an exact 10% tax would be \$0.56, and the final price after adding the rounded tax of \$0.60 should be \$6.20.

1.3 Change History

Version	Date	Change Details	Author
1.0.0	07/08/2021	The application is created as a console application, taking a txt file as input with a list of items in Json format and output to the console the calculated receipt with Sales Taxes and Total.	Apolo Orihuela

Section 2 - Overview

2.1 Purpose

Create a console application with C# that can be used to load a shopping basket items and output the receipt, also a CalculateTotal() method should be created to be used for other classes, it will receive an object (ShoppingBasket) as input and will return a receipt object as response (ShoppingBasketResponse), create a PrintReceipt() method to print a ShoppingBasketResponse in the given format.

2.2 Scope

The console application will load a txt file (ShoppingCart.txt) in Json format or receive an object (ShoppingBasket) in a proper format, the program will validate a correct Json format and will calculate the totals according to the business rules explained in the point 1.2 (description), these are the models used for this project:

```

11 references
public class ShoppingBasket
{
    24 references | 4/4 passing
    public List<Item> ItemList { get; set; } = new List<Item>();
}

30 references
public class Item
{
    4 references
    public string productDescription { get; set; }
    3 references
    public int quantity { get; set; }
    3 references
    public float price { get; set; }
    2 references
    public string typeOfProduct { get; set; }
    3 references
    public bool isImported { get; set; }
}

10 references
public class ShoppingBasketResponse
{
    7 references
    public List<ItemResponse> ItemList { get; set; } = new List<ItemResponse>();
    7 references | 4/4 passing
    public float salesTaxes { get; set; }
    7 references | 4/4 passing
    public float total { get; set; }
}

8 references
public class ItemResponse
{
    7 references
    public string productDescription { get; set; }
    7 references
    public float subtotal { get; set; }
    12 references
    public int quantity { get; set; }
    11 references
    public float unitPrice { get; set; }
    1 reference
    public float basicSalesTax { get; set; } = 0; // Is initialize with 0 in case we don't apply any tax for this item
    1 reference
    public float importTax { get; set; } = 0; // Is initialize with 0 in case we don't apply any tax for this item
    5 references
    public string typeOfProduct { get; set; }
    6 references
    public bool isImported { get; set; }
}

```

Where the response object (ShoppingBasketResponse) contains a list of ItemResponse, and each will have the shown properties, also in the response Object (ShoppingBasketResponse) will have the Sales Taxes and Total as a property.

Note: the unit price (in case quantity is greater than 1 and if any tax applies this will be added to the unit price) the basic sales tax and import tax of that item are calculated for a single item.

Example of Input Json Format:

Item

```
{
  "productDescription": "Book",
  "quantity": 1,
  "price": 12.49,
  "typeOfProduct": "book",
  "isImported": false
}
```

Shopping Basket

```
{
  "ItemList": [
    {
      "productDescription": "Book",
      "quantity": 1,
      "price": 12.49,
      "typeOfProduct": "book",
      "isImported": false
    },
    {
      "productDescription": "Book",
      "quantity": 1,
      "price": 12.49,
      "typeOfProduct": "book",
      "isImported": false
    },
    {
      "productDescription": "Music CD",
      "quantity": 1,
      "price": 14.99,
      "typeOfProduct": "music",
      "isImported": false
    },
    {
      "productDescription": "Chocolate bar",
      "quantity": 1,
      "price": 0.85,
      "typeOfProduct": "book",
      "isImported": false
    }
  ]
}
```

2.3 Requirements

1. - Write an application that takes input for shopping baskets and returns receipts in the format shown below:

INPUT 1

1 Book at 12.49
1 Book at 12.49
1 Music CD at 14.99
1 Chocolate bar at 0.85

OUTPUT 1

Book: 24.98 (2 @ 12.49)
Music CD: 16.49
Chocolate bar: 0.85
Sales Taxes: 1.50
Total: 42.32

INPUT 2

1 Imported box of chocolates at 10.00
1 Imported bottle of perfume at 47.50

OUTPUT 2

Imported box of chocolates: 10.50
Imported bottle of perfume: 54.65
Sales Taxes: 7.65
Total: 65.15

INPUT 3

1 Imported bottle of perfume at 27.99
1 Bottle of perfume at 18.99
1 Packet of headache pills at 9.75
1 Imported box of chocolates at 11.25
1 Imported box of chocolates at 11.25

OUTPUT 3

Imported bottle of perfume: 32.19
Bottle of perfume: 20.89
Packet of headache pills: 9.75
Imported box of chocolates: 23.70 (2 @ 11.85)
Sales Taxes: 7.30
Total: 86.53

2.- Calculating all taxes and totals correctly.

3. - When calculating the sales tax, round the value up to the nearest 5 cents. For example, if a taxable item costs \$5.60, an exact 10% tax would be \$0.56, and the final price after adding the rounded tax of \$0.60 should be \$6.20.

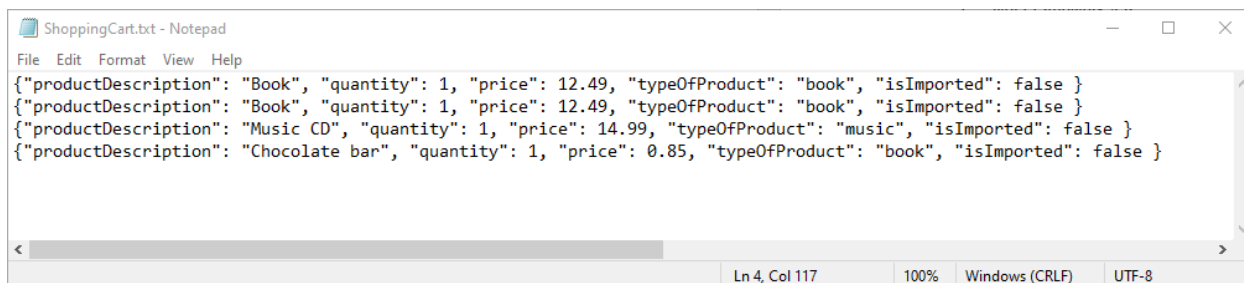
2.4 System Requirements

1. - .Net Framework 4.5
2. - Visual Studio 2019 (for deploying and testing purposes)
- 3.- Windows Operating System

2.5 Debug Mode Notes

The project was created on Visual Studio 2019, with target framework 4.5, you can safely debug the solution and set breakpoints, is recommended to change the value of the variable named "ShoppingCartTextFile" to the correct location of the text file to be used (ShoppingCart.txt) containing the Json format of the input Shopping Basket.

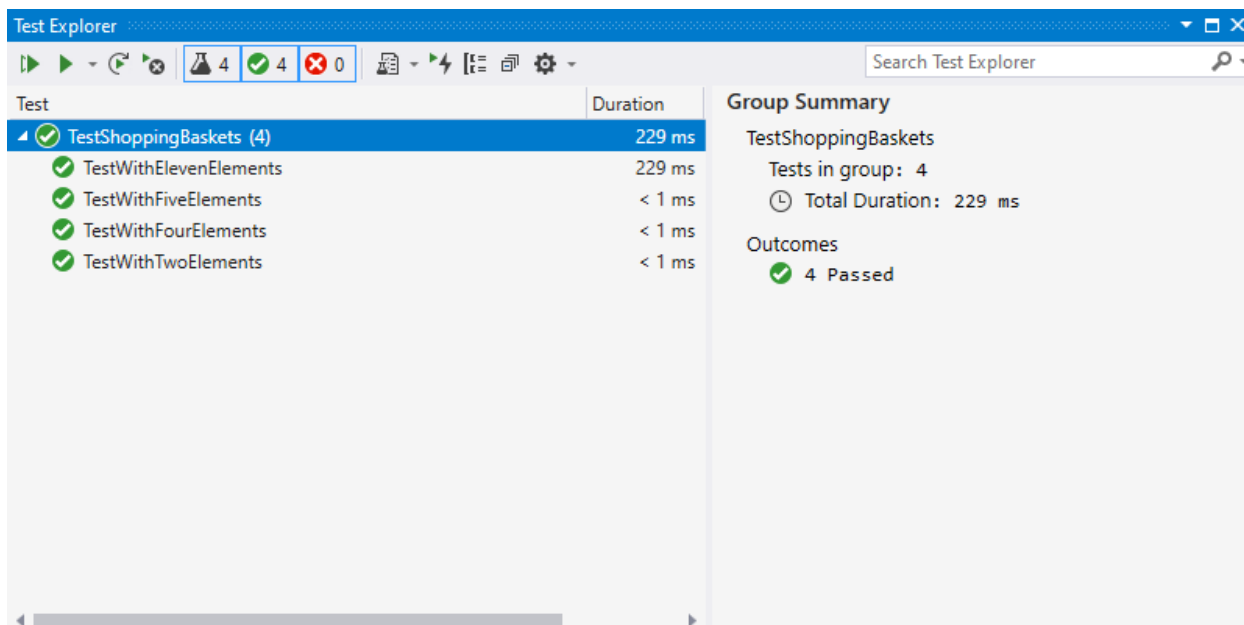
Example of ShoppingCart.txt:



```
ShoppingCart.txt - Notepad
File Edit Format View Help
{"productDescription": "Book", "quantity": 1, "price": 12.49, "typeOfProduct": "book", "isImported": false }
{"productDescription": "Book", "quantity": 1, "price": 12.49, "typeOfProduct": "book", "isImported": false }
{"productDescription": "Music CD", "quantity": 1, "price": 14.99, "typeOfProduct": "music", "isImported": false }
{"productDescription": "Chocolate bar", "quantity": 1, "price": 0.85, "typeOfProduct": "book", "isImported": false }
Ln 4, Col 117 100% Windows (CRLF) UTF-8
```

A test project (MS Unit Test) has been included with the name SalesTaxesTests, it has 4 tests Methods to test, the logic for these is simple, create a ShoppingBasket Object and process it using the CalculateTotal() Method to get the ShoppingBasketResponse Object, then compare the output to the expected output and comparing the Sales Taxes and Total against the same properties in the response object..

Example of the tests



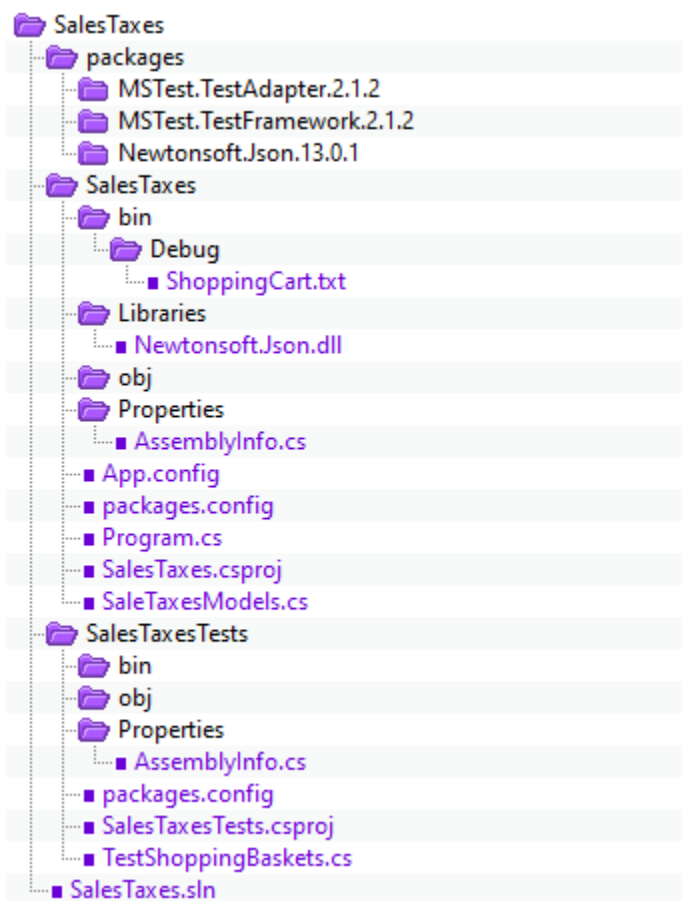
Test	Duration
TestShoppingBaskets (4)	229 ms
TestWithElevenElements	229 ms
TestWithFiveElements	< 1 ms
TestWithFourElements	< 1 ms
TestWithTwoElements	< 1 ms

Group Summary
 TestShoppingBaskets
 Tests in group: 4
 Total Duration: 229 ms
Outcomes
 4 Passed

Note: an external library is used the "Json.Net" from Newtonsoft that is added using the NuGet packages but is included in the project as an external library and is embedded inside of the deliverable, this to avoid to export it to the deployment folder and be installed in the target machine.

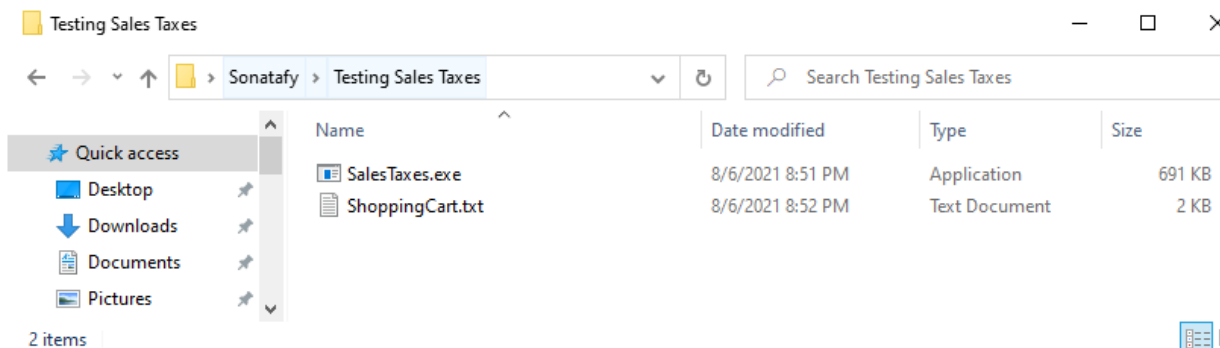
Section 3 - Deployment

3.1 File System Structure



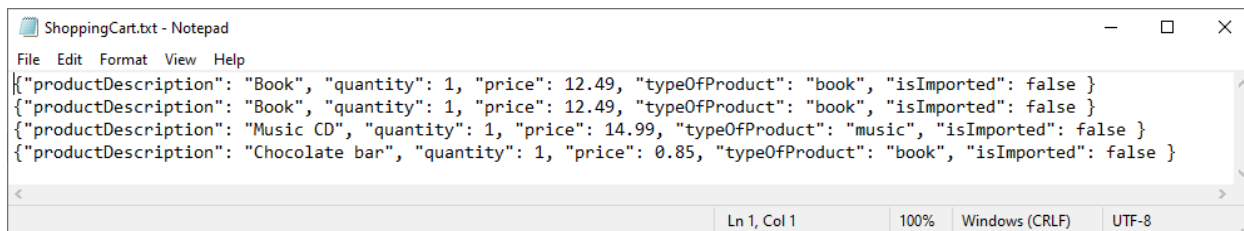
3.2 Console Application Deployment

To deploy this console application, you must open the solution file in Visual Studio 2019 in debug mode in order to get the executable file for deploying, once you have test it you can change the deployment type to Release mode. Once the exe is generated you only need to place the executable in the folder you want to use it (all necessary external libraries will be embedded inside of the EXE file). Example of deployment:



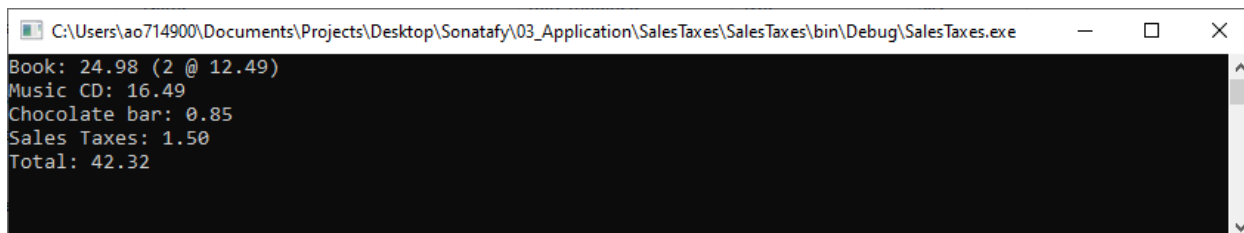
3.3 Console Application Running Examples

Test 1 Input file



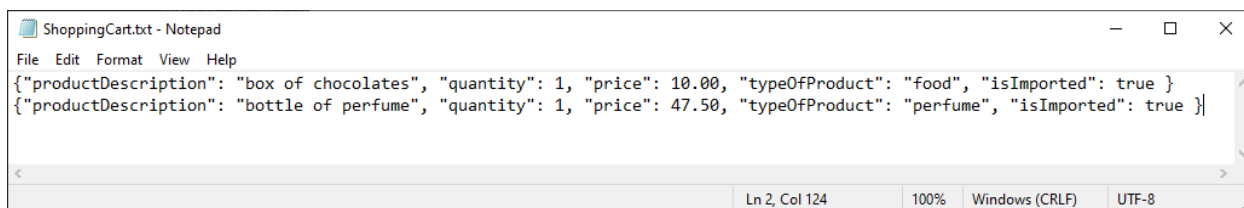
```
ShoppingCart.txt - Notepad
File Edit Format View Help
{"productDescription": "Book", "quantity": 1, "price": 12.49, "typeOfProduct": "book", "isImported": false }
{"productDescription": "Book", "quantity": 1, "price": 12.49, "typeOfProduct": "book", "isImported": false }
{"productDescription": "Music CD", "quantity": 1, "price": 14.99, "typeOfProduct": "music", "isImported": false }
{"productDescription": "Chocolate bar", "quantity": 1, "price": 0.85, "typeOfProduct": "book", "isImported": false }
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Test 1 Output



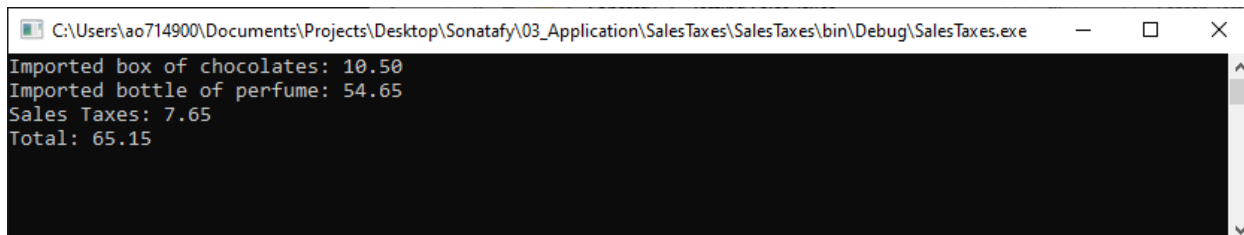
```
C:\Users\ao714900\Documents\Projects\Desktop\Sonatafy\03_Application\SalesTaxes\SalesTaxes\bin\Debug\SalesTaxes.exe
Book: 24.98 (2 @ 12.49)
Music CD: 16.49
Chocolate bar: 0.85
Sales Taxes: 1.50
Total: 42.32
```

Test 2 Input File



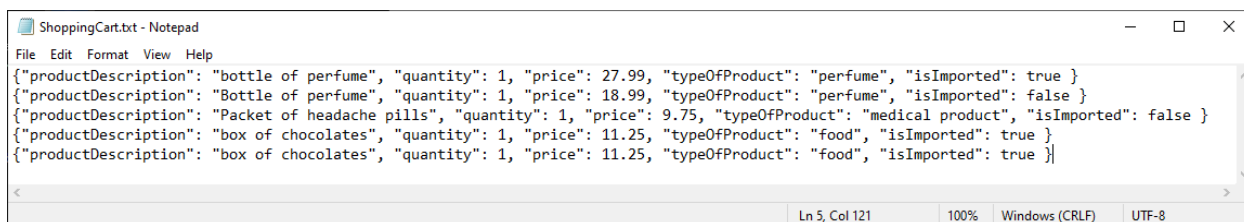
```
ShoppingCart.txt - Notepad
File Edit Format View Help
{"productDescription": "box of chocolates", "quantity": 1, "price": 10.00, "typeOfProduct": "food", "isImported": true }
{"productDescription": "bottle of perfume", "quantity": 1, "price": 47.50, "typeOfProduct": "perfume", "isImported": true }
Ln 2, Col 124 100% Windows (CRLF) UTF-8
```

Test 2 Output



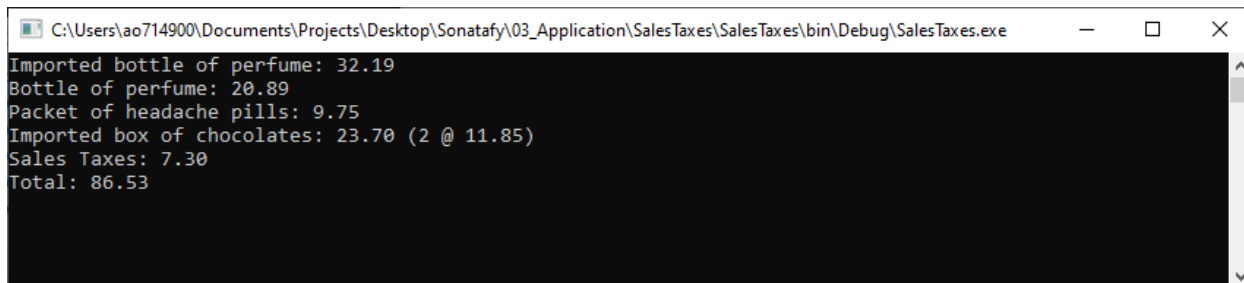
```
C:\Users\ao714900\Documents\Projects\Desktop\Sonatafy\03_Application\SalesTaxes\SalesTaxes\bin\Debug\SalesTaxes.exe
Imported box of chocolates: 10.50
Imported bottle of perfume: 54.65
Sales Taxes: 7.65
Total: 65.15
```

Test 3 Input File



```
ShoppingCart.txt - Notepad
File Edit Format View Help
{"productDescription": "bottle of perfume", "quantity": 1, "price": 27.99, "typeOfProduct": "perfume", "isImported": true }
{"productDescription": "Bottle of perfume", "quantity": 1, "price": 18.99, "typeOfProduct": "perfume", "isImported": false }
{"productDescription": "Packet of headache pills", "quantity": 1, "price": 9.75, "typeOfProduct": "medical product", "isImported": false }
{"productDescription": "box of chocolates", "quantity": 1, "price": 11.25, "typeOfProduct": "food", "isImported": true }
{"productDescription": "box of chocolates", "quantity": 1, "price": 11.25, "typeOfProduct": "food", "isImported": true }
Ln 5, Col 121 100% Windows (CRLF) UTF-8
```

Test 3 Output File



```
C:\Users\ao714900\Documents\Projects\Desktop\Sonatafy\03_Application\SalesTaxes\SalesTaxes\bin\Debug\SalesTaxes.exe
Imported bottle of perfume: 32.19
Bottle of perfume: 20.89
Packet of headache pills: 9.75
Imported box of chocolates: 23.70 (2 @ 11.85)
Sales Taxes: 7.30
Total: 86.53
```