Exam –Retail Store

Description

Create a Receipt class that could be used by a retail store. Items to include as data members are **receipt number, date of purchase, customer number, customer name and address, customer phone number, item number, description, unit price**, and **quantity purchased**.
Choose data types that are appropriate for each field. Create private class variables and the necessary Properties.

Constructor:

A constructor should be created that takes in all of the class members and sets them to the appropriate properties. Do not create any other constructors.

Property validation:

In the Set for each of the Properties perform this validation to ensure that improper values are not being set.

- Receipt number should be greater than zero
- Customer number should be greater than zero
- Item number should be greater than zero and less than 9999
- Unit price should be greater than zero and less than 9999
- Quantity purchased should be greater than zero

For simplicity, you may assume each receipt is for a single ticket item and contains a single item number.

Methods:

- Create a method that calculates the total cost using the quantity and unit price.
- Override the ToString ( ) method to return the information about the customer (name and phone number) and the total cost of the item purchased so that it appears this way –

Customer {customer first name} {customer last name}
Phone: {customer phone number}
Total Purchases: {calculated total cost}

Use monetary formatting when displaying the total cost

Testing

Create a Unit Test class that contains test methods which are testing for the following:

- Each of the 5 validation items should be tested with individual test methods. Create a test method that shows a valid input and an invalid input. (You'll have 10 test methods)