|  |  |
| --- | --- |
| Specifications | Test Case |
| Create a program that creates several Abstract Data types/ADTs(Product, Sale Item, and Sales Receipt). | Test Case 1-3: Output each ADT to satisfactorily demonstrate the data storage to the customer. |
| Store the data from the product and sales data into the appropriate ADTs. | Test Case 1-3: Output each ADT to satisfactorily demonstrate the data storage to the customer. |
| Design the capacity to print each individual ADT, with a method from within the ADT. | Test Case 1-3: Output each ADT to satisfactorily demonstrate the data storage to the customer. |
| Print from the ADTs a complete sales inventory that lists the product ID, description, unit price of each object, the number of units sold, and the total revenue from each product. | Test Case 4: Output a sales inventory that accurately matches the totals of the sales data. |

|  |
| --- |
| salesReceipt |
| -saleID int  -customerID int  -saledateTime int  -items saleItem[] |
| +receiptConstructor  +inventoryTracker()  +getSaleID  +getCustomerID  +getDateTime  +Create a linked array of saleItems[]  +calcReceiptTotal  +printReceipt |

|  |
| --- |
| saleItem |
| -Productid Int  -unitPrice Double  -units Int |
| +itemConstructor()  +setProductID(int idIn)  +setUnitPrice(double priceSet)  +setUnits(int unitsIn)  +getProductID(){query}  +getUnitPrice(){query}  +getUnits(){query}  +calcTotal()  +printSaleItem() |

|  |
| --- |
| Product |
| -productID Int  -description string  -price Int |
| +productConstructor()  +setDescription(string descriptionIn)  +setUnitPrice(int priceIn)  +getDescription() {query}  +getUnitPrice() {query}  +printProduct() {query/Print} |

Psuedo Code:

Main

Define Variables

-File Locations

-Date/Time

-Classes

Import Product Data

Store items as a string array

Generate each product class

-Parse ints into the various fields

-Store strings

Print Test Case 1: Procut Print

Generate the linkedArray to store the overall sales receipts ADTs, by scaning SalesData Array for like customer IDs

Store each item type of sale to the customer as a salesItem

Print Test Case 2

Generate an additional receipt that is created from reading the other sales receipts, and calculating the totals.

Scan the sales receipts for like product IDs

generate sums for each prduct

calaculate the totals that are presents for each product, and store the total sales unit for each item as I go into a multidimensional linked array aka sort algo.

Print Test Case 3

Class Product

Define the variables and types

Define the methods for the class

Class salesItem

Define the variables and types

Define the methods for the class

Class salesReceipt

Define the variables and types

Define the methods for the class

Class salesInventory

Define the variables and types

Define the methods for the class