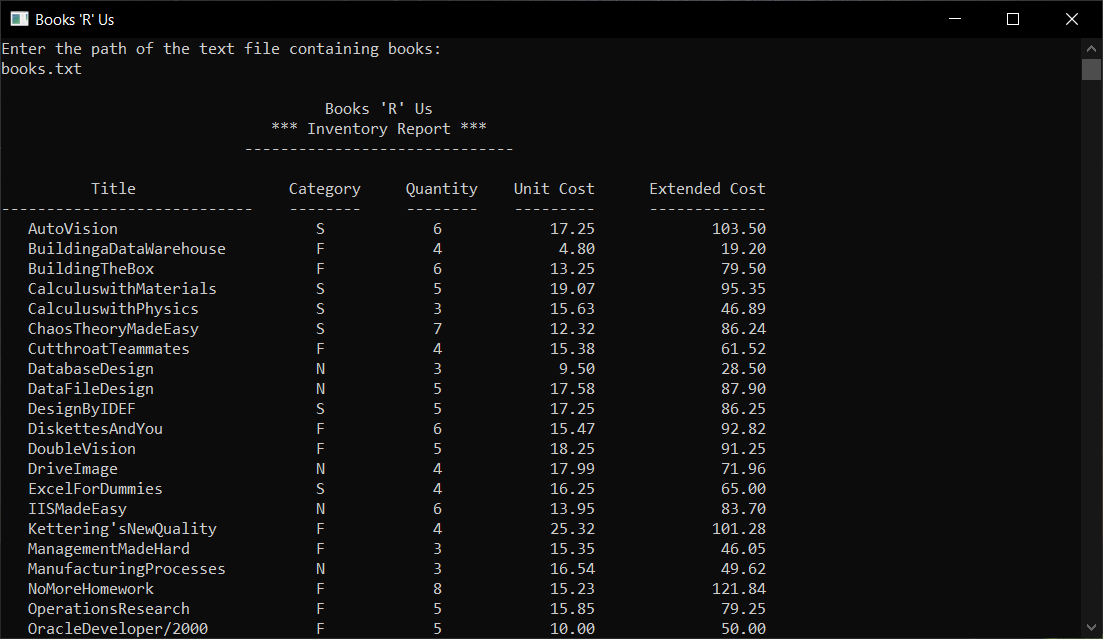
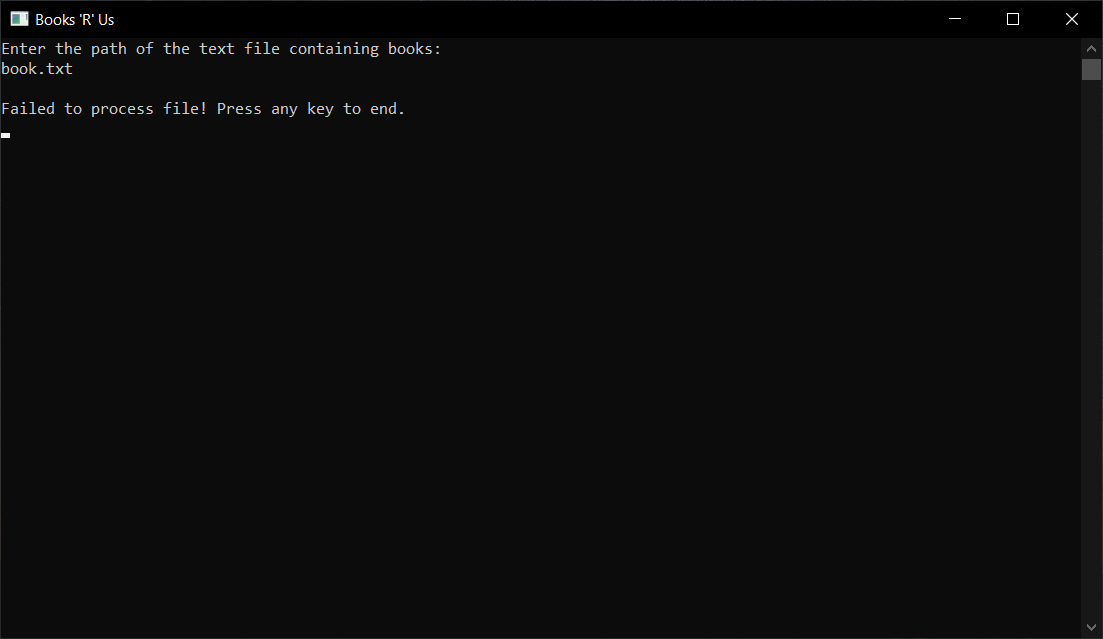
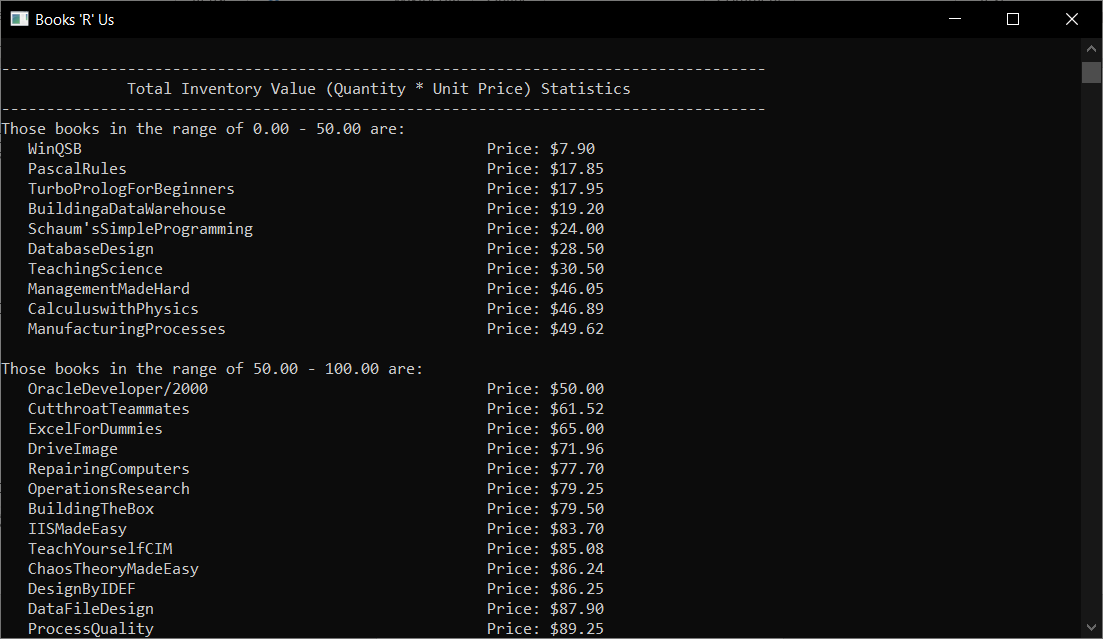
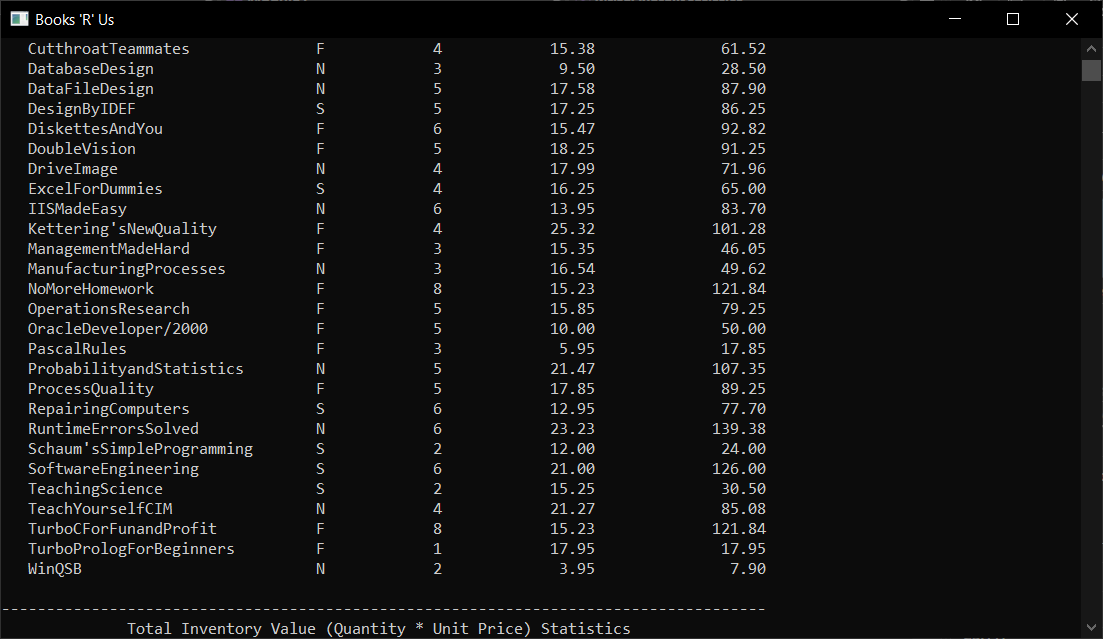
Program Cover Sheet

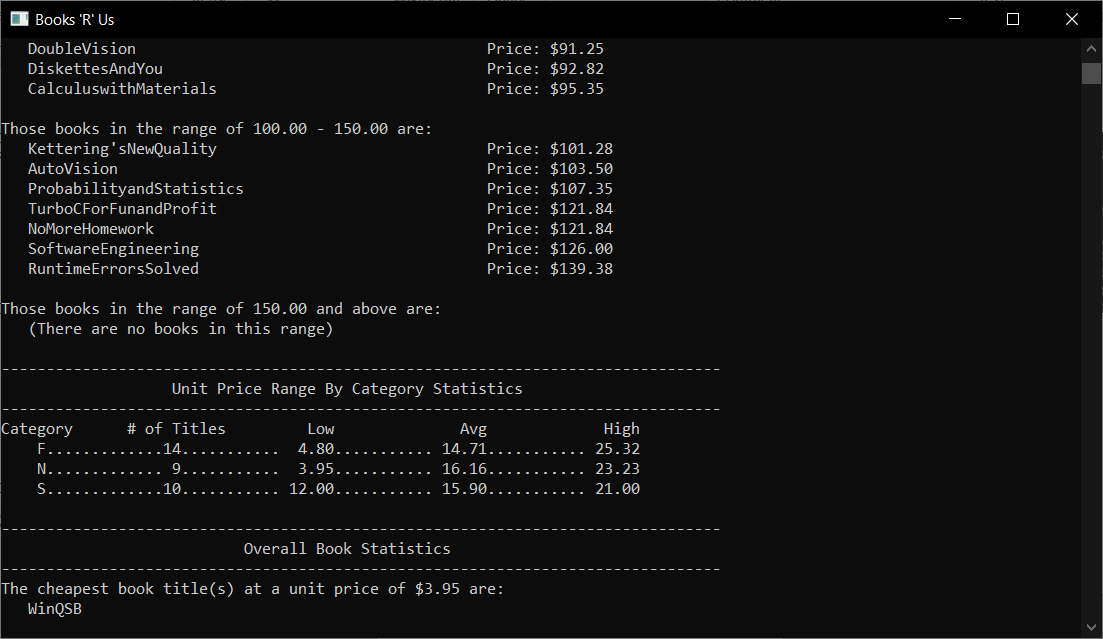
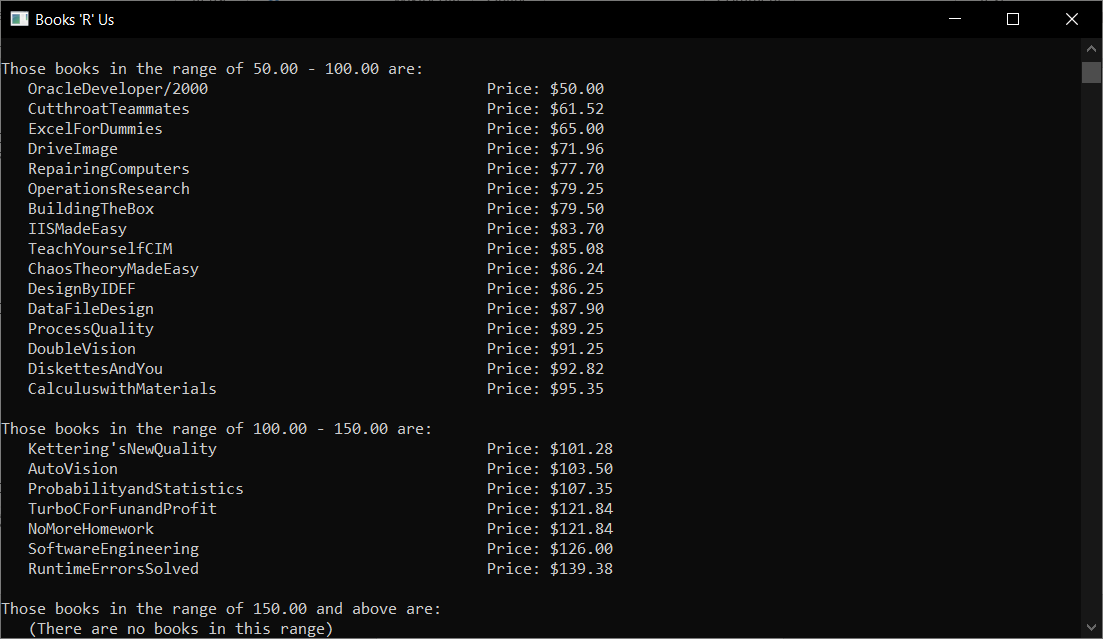
|  |
| --- |
| Name: Nathan VanSnepson |
| Assignment: Assignment 5 |
| List any parts of the assignment that do not work/were not completed: (None) |

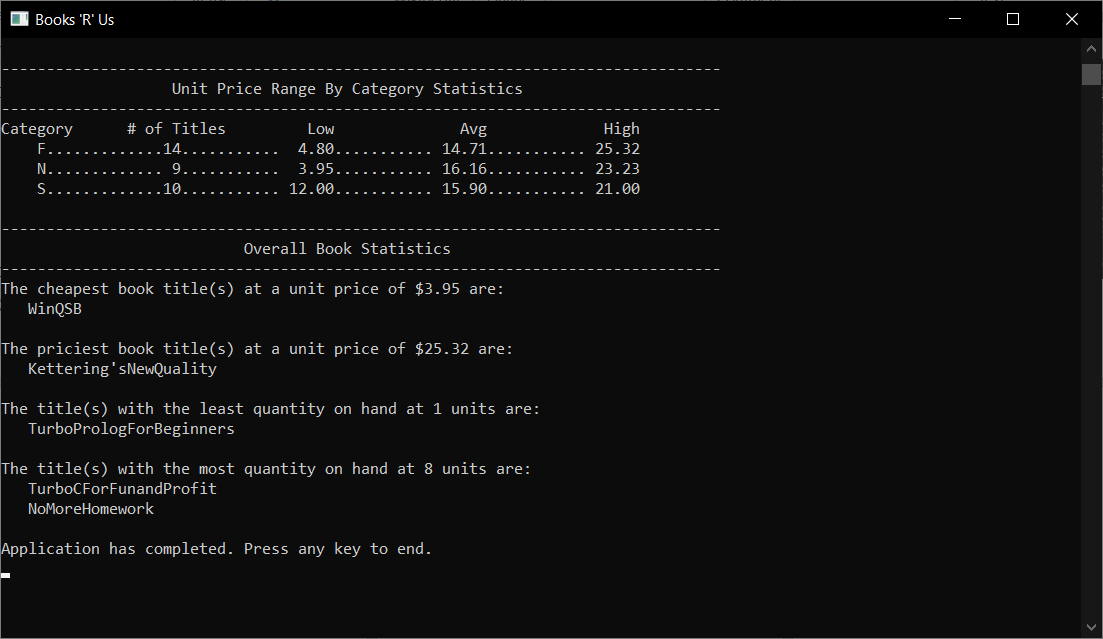
|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.









'------------------------------------------------------------

'- File Name : Book.vb -

'- Part of Project: PayrollSystem -

'------------------------------------------------------------

' Written By: Nathan VanSnepson -

' Written On: February 28, 2022 -

'------------------------------------------------------------

'- File Purpose: -

'- -

'- This file contains information that represents a book -

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'-\*intQuantity - The number of books available -

'-\*sngInventoryTotal - The total cost of all the books in -

'- the inventory. -

'-\*sngPrice - The price for a single book -

'-\*strCaregory - The genre of the book. -

'-\*strTitle - The title of the book

'------------------------------------------------------------

Public Class Book

'---------------------------------------------------------------------------------------

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'---------------------------------------------------------------------------------------

'-------------------------------------------------------------------------------------------

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'-------------------------------------------------------------------------------------------

'---------------------------------------------------------------------------------------

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'---------------------------------------------------------------------------------------

Public intQuantity As Integer

Public sngInventoryTotal As Single

Public sngPrice As Single

Public strCategory As String

Public strTitle As String

'-----------------------------------------------------------------------------------

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'-----------------------------------------------------------------------------------

Public Sub New(intQuantity As Integer, sngInventoryTotal As Single, sngPrice As Single, strCategory As String, strTitle As String)

'------------------------------------------------------------

'- Subprogram Name: new -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram is a constructor for the book class -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*intQuantity - the number of the book available -

'-\*sngInventoryTotal - the total cost quantity of book -

'- available -

'-\*sngPrice - The price per single book -

'-\*strCategory - The genre of the book -

'-strTitle - Title of the book -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*bookTemp - an instance of the book class that gets -

'- reused to add new books to the book list -

'-\*sbBookName - A StringBuilder to add the book title -

'-\*srReadInputFile - StreamReader to read in from a file -

'-\*strInputStream - String array that holds the value of a -

'- line from the file -

'------------------------------------------------------------

Me.intQuantity = intQuantity

Me.sngInventoryTotal = sngInventoryTotal

Me.sngPrice = sngPrice

Me.strCategory = strCategory

Me.strTitle = strTitle

End Sub

End Class

'------------------------------------------------------------

'- File Name : Module1.vb -

'- Part of Project: Assignment5 -

'------------------------------------------------------------

' Written By: Nathan VanSnepson -

' Written On: February 28, 2022 -

'------------------------------------------------------------

'- File Purpose: -

'- -

'- This file contains the main program for outputing the -

'- statistics of a file of books. -

'------------------------------------------------------------

'- Program Purpose: -

'- -

'- The purpose of this program is to read in a text file of -

'- books and write the statistics out to the user. -

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'- intEmployeeAccessed - The Position in the udtEmployeeList-

'- where the currently displayed employee is located. -

'- udtEmployeeList - ArrayList containing Employee informa- -

'- tion

'------------------------------------------------------------

Imports System.IO

Imports System.Text

Module Module1

'---------------------------------------------------------------------------------------

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'---------------------------------------------------------------------------------------

Const COMMAND\_WIDTH = 120 'Width of the command prompt

Const COMMAND\_HEIGHT = 30 'Height of the command prompt

Const COMMMAND\_TITLE = "Books 'R' Us" 'Title of the command prompt

'-------------------------------------------------------------------------------------------

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'-------------------------------------------------------------------------------------------

'---------------------------------------------------------------------------------------

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'---------------------------------------------------------------------------------------

'-----------------------------------------------------------------------------------

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'-----------------------------------------------------------------------------------

Sub Main()

'------------------------------------------------------------

'- Subprogram Name: Main -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to read from aa book -

'- text file and display statistics of the books. This -

'- subprogram handles exiting the program early if there is -

'- an error.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*lstBooks - a list containing book objects -

'-\*strFileName - String of the fileName for the book text -

'- file.

'------------------------------------------------------------

Dim lstBooks As New List(Of Book) 'List of books

Dim strFileName As String 'Books file name

'Customizes the design of the console (e.g. size)

customizeConsole()

'Prompts user to enter the filepath of the book text file

Console.WriteLine("Enter the path of the text file containing books: ")

strFileName = Console.ReadLine()

Console.WriteLine()

'Reads the file and inserts books into lstBooks, if there is an error

'display why to the user then exits the program

If Not processFile(strFileName, lstBooks) Then

endProgram("Failed to process file! ")

Return

End If

'Writes the books to console in alphabetic order by books Title, if there is an error

'display why to the user then exits the program

If Not writeBooksToConsole(lstBooks) Then

endProgram("Failed to write books to console! ")

Return

End If

'Write the inventory statistics of the books, if there is an error

'display why to the user then exits the program

If Not writeInventoryStatistics(lstBooks) Then

endProgram("Failed to write inventory statistics to console! ")

Return

End If

'Writes the category statistics of the books, if there is an error

'display why to the user then exits the program

If Not writeCategoryStatistics(lstBooks) Then

endProgram("Failed to write category statistics to console! ")

Return

End If

'Writes the Overall statisctics of the books, if there is an error

'display why to the user then exits the program

If Not writeOverallStatistics(lstBooks) Then

endProgram("Failed to write overall book statistics to console! ")

Return

End If

'Prompt user the application has completed successfully

endProgram("Application has completed. ")

End Sub

Sub customizeConsole()

'------------------------------------------------------------

'- Subprogram Name: customizeConsole -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram customizes the Conosole to a specfied -

'- design. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (none)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (none) -

'------------------------------------------------------------

Console.Title = COMMMAND\_TITLE

Console.SetWindowSize(COMMAND\_WIDTH, COMMAND\_HEIGHT)

Console.Clear()

End Sub

Sub endProgram(strMessage As String)

'------------------------------------------------------------

'- Subprogram Name: endProgram -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 27, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram is called when the program wants to exit -

'- and informs the user why the program is quitting. It -

'- asks user to hit any key to quit. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*strMessage - String containing the quit message to output-

'- to the user. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (none) -

'------------------------------------------------------------

Console.Write(strMessage)

Console.WriteLine("Press any key to end.")

Console.ReadKey()

End Sub

Function processFile(ByVal strFileName As String, ByRef lstBooks As List(Of Book))

'------------------------------------------------------------

'- Subprogram Name: processFile -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram reads in from a file and writes the data -

'- to a book list. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*strFileName - Name of the file to read -

'-\*lstBooks - Referance to a book list -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*bookTemp - an instance of the book class that gets -

'- reused to add new books to the book list -

'-\*sbBookName - A StringBuilder to add the book title -

'-\*srReadInputFile - StreamReader to read in from a file -

'-\*strInputStream - String array that holds the value of a -

'- line from the file -

'------------------------------------------------------------

Dim bookTemp As Book 'Instance of the book class that gets reused to add new books to the book list

Dim sbBookName As StringBuilder 'A StringBuilder to add the book title

Dim srReadInputFile As StreamReader 'StreamReader to read in from a file

Dim strInputStream As String() 'String array that holds the value of a line from the file

'Try statement to catch any encountered errors

Try

'Checks if the file exists

If File.Exists(strFileName) Then

'Creates a new instance of the StreamReader to read in from the file

srReadInputFile = New StreamReader(strFileName)

'While loop to read through the file

While Not srReadInputFile.EndOfStream

strInputStream = srReadInputFile.ReadLine().Split(" ") 'String array that equals the line from the file split on spaces

sbBookName = New StringBuilder 'A new instance of the StringBuilder

'Loops through array to add the book table together

For i As Integer = 3 To strInputStream.Length - 1 Step 1

sbBookName.Append(strInputStream(i))

Next

'Creates a new instance of the book

bookTemp = New Book(

Integer.Parse(strInputStream(1)),

Double.Parse(strInputStream(2)) \* Integer.Parse(strInputStream(1)),

Double.Parse(strInputStream(2)),

strInputStream(0),

sbBookName.ToString()

)

'Adds book to the book list

lstBooks.Add(bookTemp)

End While

Return True 'Successfully exits the sub program

End If

Catch ex As Exception

Console.WriteLine(ex.Message)

Return False 'Unsuccessfully exits the sub program

End Try

Return False 'Unsuccessfully exits the sub program

End Function

Function writeBooksToConsole(ByRef lstBooks As List(Of Book))

'------------------------------------------------------------

'- Subprogram Name: writeBooksToConsole -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram writes the books in the book list to the -

'- console in alphabetical title order -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*lstBooks - Referance to a book list -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*LINQResults - Object that will contain the results of a -

'- LINQ statement. -

'------------------------------------------------------------

Dim LINQResults As Object 'Object that will contain the results of a LINQ statement

'Try statement to catch any encountered errors

Try

'Writes header of report to console

Console.WriteLine(StrDup(36, " ") & "Books 'R' Us")

Console.WriteLine(StrDup(30, " ") & "\*\*\* Inventory Report \*\*\*")

Console.WriteLine(StrDup(27, " ") & StrDup(30, "-"))

Console.WriteLine()

'Writes line to console with headers

Console.WriteLine("{0, 15} {1, 24} {2, 12} {3, 12} {4, 18}",

"Title", "Category", "Quantity", "Unit Cost", "Extended Cost")

'Dash lines under headers

Console.WriteLine("{0, -28} {1, 11} {2, 12} {3, 12} {4, 18}",

StrDup(28, "-"), StrDup(8, "-"), StrDup(8, "-"),

StrDup(9, "-"), StrDup(13, "-"))

'Sets the LINQResults = the lstBooks in alphabetical title order

LINQResults = (From book In lstBooks

Select book.strTitle, book.strCategory, book.intQuantity, book.sngPrice, book.sngInventoryTotal

Order By strTitle)

'Loops through the LINQResults of books and writes the books to the console

For Each book In LINQResults

Console.WriteLine(" {0, -28} {1, 4} {2, 12} {3, 16} {4, 18}",

book.strTitle, book.strCategory, book.intQuantity.ToString(),

Format(book.sngPrice, "0.00"), Format(book.sngInventoryTotal, "0.00"))

Next

Console.WriteLine()

Return True 'Successfully exits the sub program

Catch ex As Exception

Console.WriteLine(ex.Message)

Return False 'Unsuccessfully exits the sub program

End Try

Return False 'Unsuccessfully exits the sub program

End Function

Function writeInventoryStatistics(ByRef lstBooks As List(Of Book))

'------------------------------------------------------------

'- Subprogram Name: writeInventoryStatistics -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram writes the books grouped by the total -

'- cost of those books (Quantity \* Unit Price) -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*lstBooks - Referance to a book list -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Try statement to catch any encountered errors

Try

'Writes header for inventory statistics to the console

Console.WriteLine(StrDup(85, "-"))

Console.WriteLine(" Total Inventory Value (Quantity \* Unit Price) Statistics")

Console.WriteLine(StrDup(85, "-"))

'Calls method to write the books by their price range

queryInventoryRange(0, 50, lstBooks)

queryInventoryRange(50, 100, lstBooks)

queryInventoryRange(100, 150, lstBooks)

queryInventoryRange(150, Single.MaxValue, lstBooks)

Return True 'Successfully exits the sub program

Catch ex As Exception

Console.WriteLine(ex.Message)

Return False 'Unsuccessfully exits the sub program

End Try

Return False 'Unsuccessfully exits the sub program

End Function

Sub queryInventoryRange(sngLow As Single, sngHigh As Single, ByRef lstBooks As List(Of Book))

'------------------------------------------------------------

'- Subprogram Name: queryInventoryRange -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram writes the books grouped between the high-

'- and low cost. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*sngLow - The lowest cost of the books range -

'-\*sngHigh - The highest cost of the book range -

'-\*lstBooks - Referance to a book list -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*LINQResults - Object that will contain the results of a -

'- LINQ statement. -

'------------------------------------------------------------

Dim LINQResults As Object 'Object that will contain the results of a LINQ statement

'Try statement to catch any encountered errors

Try

'Statement to get a result from LINQ depending on if there are any books that are between

'The minimum and maximum costs

LINQResults = IIf((From book In lstBooks

Where book.sngInventoryTotal < sngHigh And book.sngInventoryTotal >= sngLow

Select book.strTitle, book.sngInventoryTotal

Order By sngInventoryTotal).Count > 0,

(From book In lstBooks

Where book.sngInventoryTotal < sngHigh And book.sngInventoryTotal >= sngLow

Select book.strTitle, book.sngInventoryTotal

Order By sngInventoryTotal),

Nothing)

'if the high is not the largest Single then write out the header for the range

If sngHigh <> Single.MaxValue Then

Console.WriteLine("Those books in the range of {0} - {1} are:",

Format(sngLow, "0.00"), Format(sngHigh, "0.00"))

Else

'Writes out the header for the range that has sngHigh as Single.MaxValue

Console.WriteLine("Those books in the range of {0} and above are:",

Format(sngLow, "0.00"))

End If

'Checks if LINQ is NOTHING

If LINQResults Is Nothing Then

'Write there are no books in the range

Console.WriteLine(" (There are no books in this range)")

Else

'Loops through the books in LINQResults and prints the books

For Each book In LINQResults

Console.WriteLine(" {0, -50} Price: {1}", book.strTitle, FormatCurrency(book.sngInventoryTotal))

Next

End If

Console.WriteLine()

Catch ex As Exception

Console.WriteLine(ex.Message)

End Try

End Sub

Function writeCategoryStatistics(lstBooks As List(Of Book))

'------------------------------------------------------------

'- Subprogram Name: writeCategoryStatistics -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram writes the category statistics of the -

'- books list -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*lstBooks - Referance to a book list -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*LINQResults - Object that will contain the results of a -

'- LINQ statement. -

'------------------------------------------------------------

Dim LINQResults As Object 'Object that will contain the results of a LINQ statement

'Try statement to catch any encountered errors

Try

'Writes the header for the category statistics

Console.WriteLine(StrDup(80, "-"))

Console.WriteLine("{0, 58}", "Unit Price Range By Category Statistics")

Console.WriteLine(StrDup(80, "-"))

Console.WriteLine("{0, 8} {1, 16} {2,11} {3,16} {4, 16}", "Category", "# of Titles", "Low", "Avg", "High")

'Gets the aggregate functions Count, Min, Max, and Avg by Category

LINQResults = (From book In lstBooks

Group book By book.strCategory

Into b = Group

Select num = b.Count(),

low = b.Min(Function(x) x.sngPrice),

high = b.Max(Function(x) x.sngPrice),

avg = b.Average(Function(x) x.sngPrice),

categorgy = strCategory

)

'Loops through each book returned and prints the category statistics

For Each book In LINQResults

Console.WriteLine(" {0, 1}{1,13}{2,2}{3,11}{4,6}{5,11}{6,6}{7,11}{8,6}",

book.categorgy, StrDup(13, "."), book.num, StrDup(11, "."),

Format(book.low, "0.00"), StrDup(11, "."), Format(book.avg, "0.00"),

StrDup(11, "."), Format(book.high, "0.00"))

Next

Console.WriteLine()

Return True 'Successfully exits the sub program

Catch ex As Exception

Console.WriteLine(ex.Message)

Return False 'Unsuccessfully exits the sub program

End Try

Return False 'Unsuccessfully exits the sub program

End Function

Function writeOverallStatistics(ByRef lstBook As List(Of Book))

'------------------------------------------------------------

'- Subprogram Name: writeOverallStatistics -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: February 28, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subprogram writes the overall statistics of the -

'- books list. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'-\*lstBooks - Referance to a book list -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'-\*intLeastQuan - The count of the book(s) with the lowest -

'- quantity. -

'-\*intMostQuan - The count of the book(s) with the most -

'- quantity. -

'-\*LINQResults - Object that will contain the results of a -

'- LINQ statement. -

'-\*sngMax - the maximum price of book(s) -

'-\*sngMin - the minimum price of book(s) -

'------------------------------------------------------------

Dim intLeastQuan As Integer 'The count of the book(s) with the lowest quantity

Dim intMostQuan As Integer 'The count of the book(s) with the most quantity

Dim LINQResults As Object 'Object that will contain the results of a LINQ statement

Dim sngMax As Single 'The maximum price of book(s)

Dim sngMin As Single 'The minimum price of book(s)

'Try statement to catch any encountered errors

Try

'Writes header for the Overall Book Statistics

Console.WriteLine(StrDup(80, "-"))

Console.WriteLine("{0, 50}", "Overall Book Statistics")

Console.WriteLine(StrDup(80, "-"))

'Gets the min price of the books

sngMin = (From book In lstBook

Select book.sngPrice

).Min()

'Gets all the books that have the min price

LINQResults = (From book In lstBook

Where book.sngPrice = sngMin

Select book)

'Writes the cheapest book price

Console.WriteLine("The cheapest book title(s) at a unit price of {0} are:", FormatCurrency(sngMin))

'Loops through the books and writes the cheapest books

For Each book In LINQResults

Console.WriteLine(" {0}", book.strTitle)

Next

Console.WriteLine()

'Gets the max price of books

sngMax = (From book In lstBook

Select book.sngPrice

).Max()

'Gets all books that have the max price

LINQResults = (From book In lstBook

Where book.sngPrice = sngMax

Select book)

'writes the max price

Console.WriteLine("The priciest book title(s) at a unit price of {0} are:", FormatCurrency(sngMax))

'Writes books that have max price

For Each book In LINQResults

Console.WriteLine(" {0}", book.strTitle)

Next

Console.WriteLine()

'Gets the least quantity of the books

intLeastQuan = (From book In lstBook

Select book.intQuantity

).Min()

'Gets the books of least quantity

LINQResults = (From book In lstBook

Where book.intQuantity = intLeastQuan

Select book)

'Write the least quantity

Console.WriteLine("The title(s) with the least quantity on hand at {0} units are:", intLeastQuan)

'Writes books of least quantity

For Each book In LINQResults

Console.WriteLine(" {0}", book.strTitle)

Next

Console.WriteLine()

'Gets the max quantity of books

intMostQuan = (From book In lstBook

Select book.intQuantity

).Max()

'Gets books that have the max quantity

LINQResults = (From book In lstBook

Where book.intQuantity = intMostQuan

Select book)

'Writes the max quantity of books

Console.WriteLine("The title(s) with the most quantity on hand at {0} units are:", intMostQuan)

'Write the books that have the max quantity

For Each book In LINQResults

Console.WriteLine(" {0}", book.strTitle)

Next

Console.WriteLine()

Return True 'Successfully exits the sub program

Catch ex As Exception

Console.WriteLine(ex.Message)

Return False 'Unsuccessfully exits the sub program

End Try

Return False 'Unsuccessfully exits the sub program

End Function

End Module