Andrew Morris | Junhee Bae | Brandon Nguyen | Christopher Banh

Team Come On and Slam, and Welcome to the Jam

Requirements Analysis Documentation

This program will practically convert Serendipity Booksellers' computer into a cash register and inventory database. The program is to perform the following tasks:
• Calculate the total of a sale, including sales tax
• When a book is purchased, subtract it from the inventory file
• Add, change, delete, and look up books in the inventory file
• Display various sales reports

The program is divided into three modules: Cashier, Inventory Database, and Report.

Main menu displays welcome screen, date, and options to launch one of the three modules or to exit the program

Inventory Database Module

The Inventory Database will be a file containing a list of all the books in Serendipity's inventory. The following information for each book will be stored in the file:

Field	**Description**
ISBN	**This is the International Standard Book Number. It is a unique number assigned to each book by the publisher.**
Title	**The title of the book.**
Author	**The book's author.**
Publisher	**The company that publishes the book.**
Date Added	**The date the book was added to the inventory.**
Quantity-On-Hand	**The number of copies of the book in inventory.**
Wholesale Cost	**The price paid by Serendipity for each copy of the book.**
Retail Price	**The price Serendipity is charging for each copy of the book.**

A separate Book class will hold all the data from above fields and a vector of objects of this class can be used as the listing of the store's inventory. Can make accessor and mutator functions or can name other classes as friend classes so as to access these private member variables from outside the class' definition.

This is the largest of the modules, and requires the implementation of many different functions not only for each of the menu options, but also for different versions of them. Inventory is a derived class of virtual class Menu. Reading the file can be a separate function from writing the file, and by using these functions we can simply organize the book list within the vector of books created in the program rather than navigating the file itself. Public functions include different functions for looking up a book by each of the book's fields because of the fact that they can be different data types.

Inventory menu needs to check all input to make sure that it's valid in type as well as range of responses/input allowed. Menu options are Look Up Book, Add Book, Delete Book, Edit Book's Record, and Return to Main Menu. At beginning of inventory function running it should open the book list file and close it at the end. To search around the book list can use markers for each location in the vector, in the form of ints. To sort the list can just use greater or less than to compare equivalent fields.

Most important part of Inventory class is being able to read and write to the list file without any errors, so that the rest of the program can access it and everything else can be written.

Cashier Module

The Cashier module allows the computer to act as a cash register. The user enters information for the books being purchased and the program calculates the sales tax and the total price. In addition, the books being purchased are automatically subtracted from the Inventory Database.

For the cashier module, we used created a class derived from the Inventory class so as to access the private member variables easily and use the functions from the Inventory class without having to create an Inventory object to call them. There are three main functions we felt the cashier module should include: the ability to add books to the list of books being sold, a function to remove them from this list, and lastly a function which displays the required calculations and removes the books from the inventory file.

The sales tax calculation is done via a constant variable saved as a private member of the Cashier class. Exception handling was included here as there was a good place to make use of it, in the calling of the addBook function. If the inventory is out of the book which the user attempts to add an exception can be thrown to tell them that there are none left. The menu uses the same switch statement format as inventory, and only three functions were used for this class, aside from the inherited functions.

Report Module

The Report module will analyze the information in the Inventory Database to produce any of the following reports:

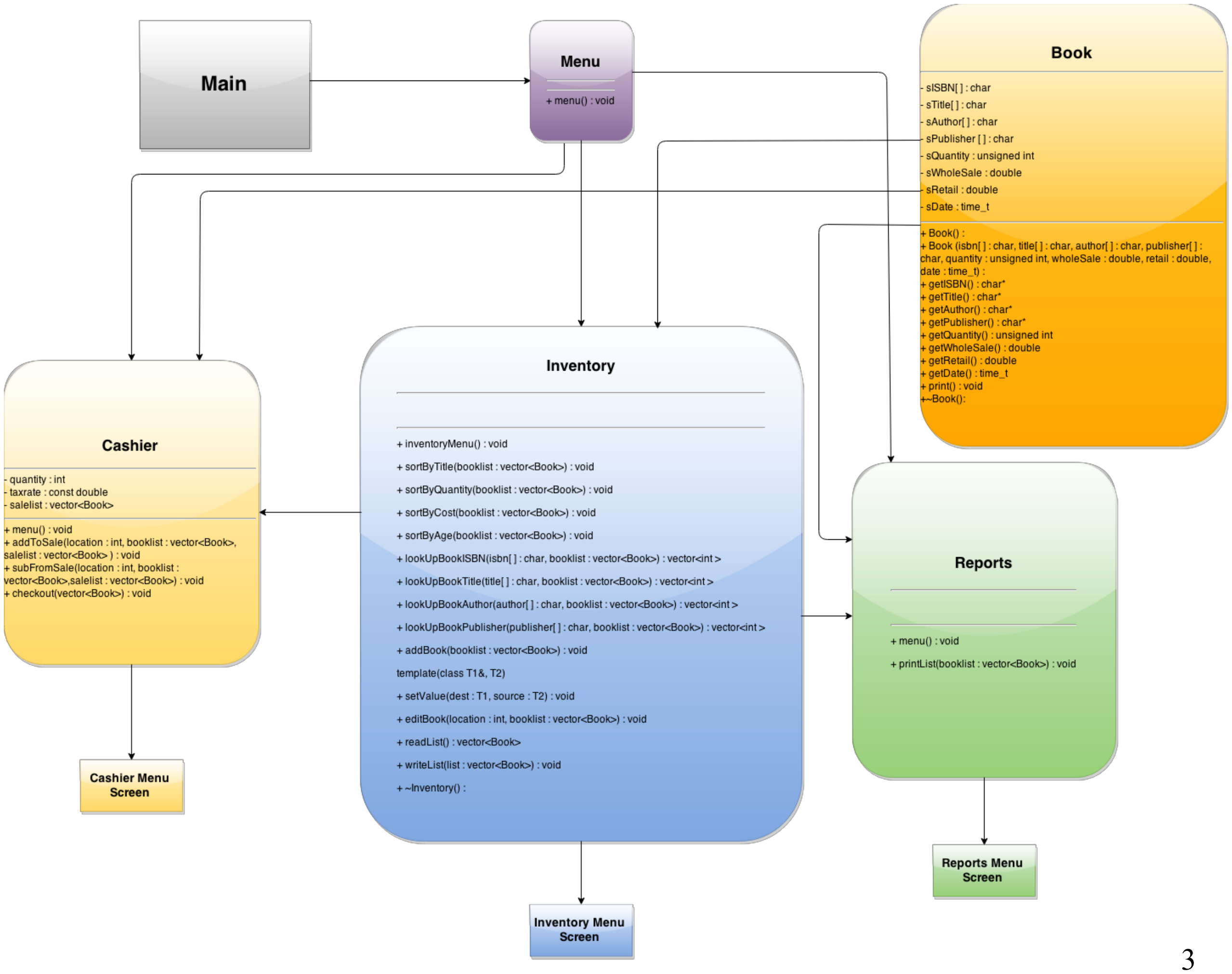
- **Inventory List: A list of information on all books in the inventory.**

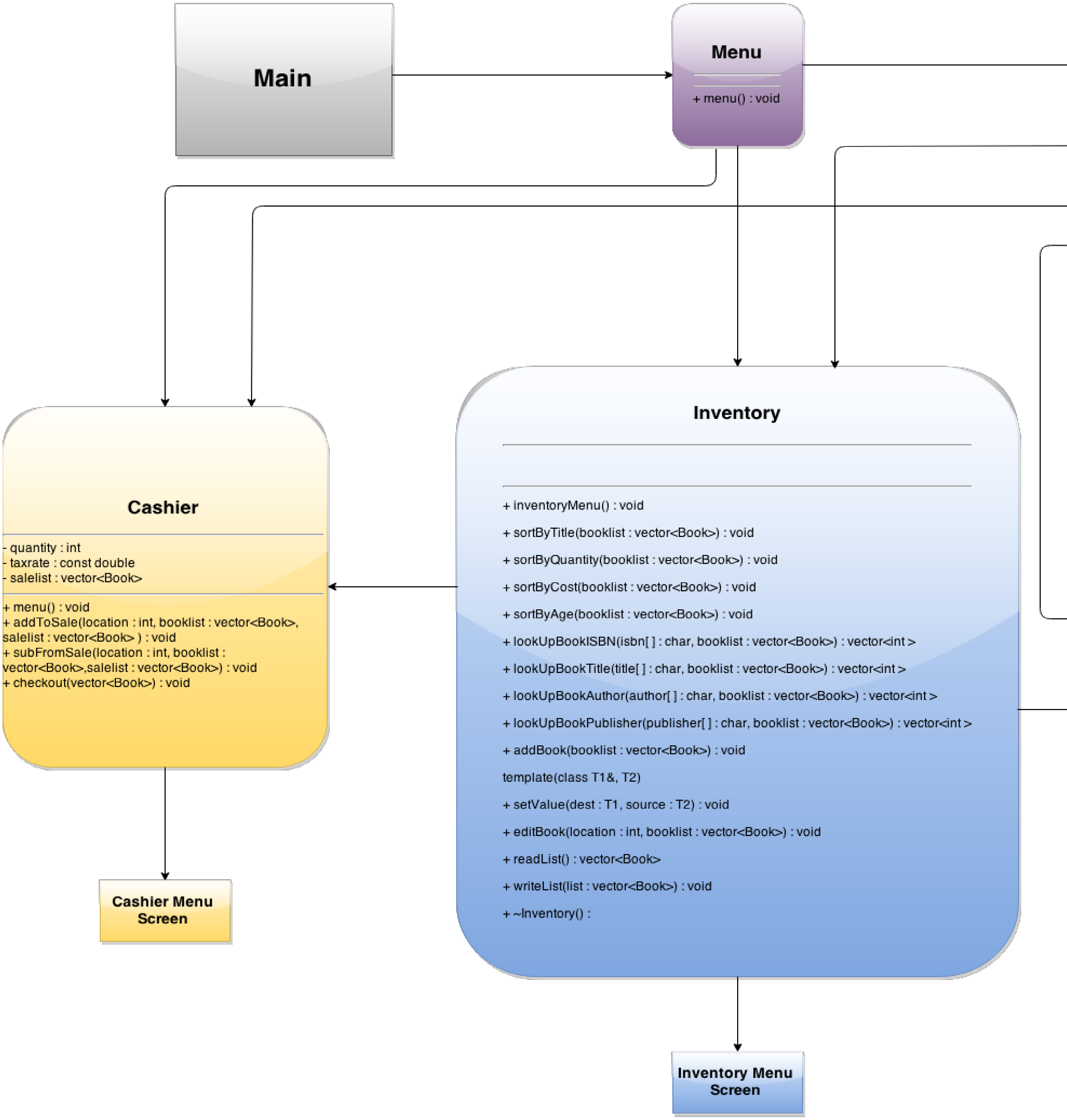
- **Inventory Wholesale Value: A list of the wholesale value of all books in the inventory and the total wholesale value of the inventory.**
- **Inventory Retail Value: A list of the retail value of all books in the inventory and the total retail value of the inventory.**
- **List by Quantity: A list of all books in the inventory sorted by quantity on hand. The books with the greatest quantity on hand will be listed first.**
- **List by Cost: A list of all books in the inventory, sorted by wholesale cost. The books with the greatest wholesale cost will be listed first.**
- **List by Age: A list of all books in the inventory, sorted by purchase date. The books that have been in the inventory longest will be listed first.**

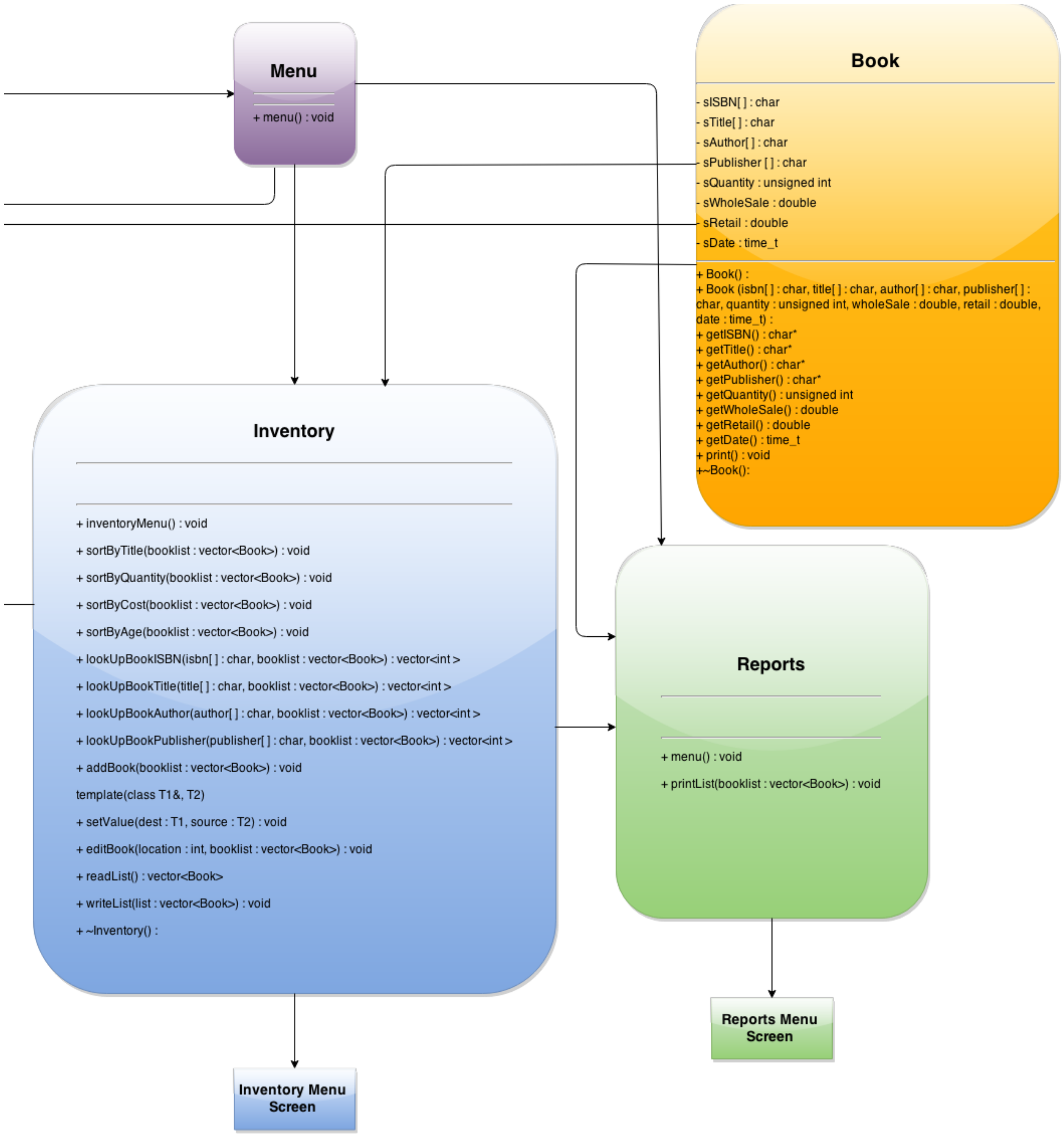
The report module is the smallest, as there is very little input and therefore we don't have to check the input constantly to make sure it's acceptable. For many of the options in this menu we can use the functions from Inventory, and if we make Report class a derived class of Inventory it will be even easier to use its functions. Probably want to use clear screen because of the long lists being displayed.

UML Diagrams:

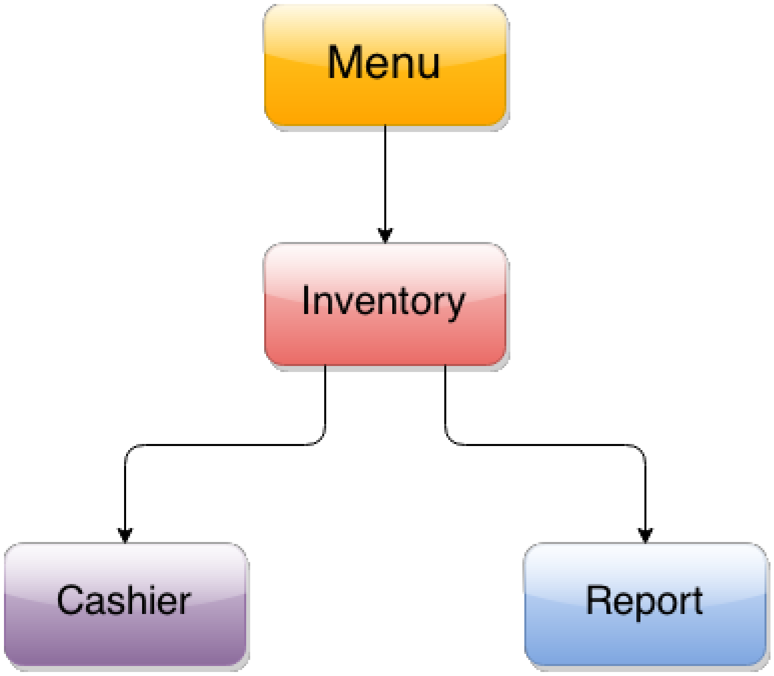
Overall Program:


Main
Menu
+ menu() : void
Book
- sISBN[ ] : char
- sTitle[ ] : char
- sAuthor[ ] : char
- sPublisher [ ] : char
- sQuantity : unsigned int
- sWholeSale : double
- sRetail : double
- sDate : time_t
+ Book() :
+ Book (isbn[ ] : char, title[ ] : char, author[ ] : char, publisher[ ] : char, quantity : unsigned int, wholeSale : double, retail : double, date : time_t) :
+ getISBN() : char*
+ getTitle() : char*
+ getAuthor() : char*
+ getPublisher() : char*
+ getQuantity() : unsigned int
+ getWholeSale() : double
+ getRetail() : double
+ getDate() : time_t
+ print() : void
+~Book():
Cashier
- quantity : int
- taxrate : const double
- salelist : vector<Book>
+ menu() : void
+ addToSale(location : int, booklist : vector<Book>, salelist : vector<Book>) : void
+ subFromSale(location : int, booklist : vector<Book>,salelist : vector<Book>) : void
+ checkout(vector<Book>) : void
Inventory
+ inventoryMenu() : void
+ sortByTitle(booklist : vector<Book>) : void
+ sortByQuantity(booklist : vector<Book>) : void
+ sortByCost(booklist : vector<Book>) : void
+ sortByAge(booklist : vector<Book>) : void
+ lookUpBookISBN(isbn[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookTitle(title[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookAuthor(author[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookPublisher(publisher[ ] : char, booklist : vector<Book>) : vector<int >
+ addBook(booklist : vector<Book>) : void
template(class T1&, T2)
+ setValue(dest : T1, source : T2) : void
+ editBook(location : int, booklist : vector<Book>) : void
+ readList() : vector<Book>
+ writeList(list : vector<Book>) : void
+ ~Inventory() :
Reports
+ menu() : void
+ printList(booklist : vector<Book>) : void
Cashier Menu Screen
Inventory Menu Screen
Reports Menu Screen


Main
Menu
+ menu() : void
Cashier
- quantity : int
- taxrate : const double
- salelist : vector<Book>
+ menu() : void
+ addToSale(location : int, booklist : vector<Book>, salelist : vector<Book>) : void
+ subFromSale(location : int, booklist : vector<Book>,salelist : vector<Book>) : void
+ checkout(vector<Book>) : void
Cashier Menu Screen
Inventory
+ inventoryMenu() : void
+ sortByTitle(booklist : vector<Book>) : void
+ sortByQuantity(booklist : vector<Book>) : void
+ sortByCost(booklist : vector<Book>) : void
+ sortByAge(booklist : vector<Book>) : void
+ lookUpBookISBN(isbn[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookTitle(title[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookAuthor(author[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookPublisher(publisher[ ] : char, booklist : vector<Book>) : vector<int >
+ addBook(booklist : vector<Book>) : void
template(class T1&, T2)
+ setValue(dest : T1, source : T2) : void
+ editBook(location : int, booklist : vector<Book>) : void
+ readList() : vector<Book>
+ writeList(list : vector<Book>) : void
+ ~Inventory() :
Inventory Menu Screen


Menu
+ menu() : void
Book
- sISBN[ ] : char
- sTitle[ ] : char
- sAuthor[ ] : char
- sPublisher [ ] : char
- sQuantity : unsigned int
- sWholeSale : double
- sRetail : double
- sDate : time_t
+ Book() :
+ Book (isbn[ ] : char, title[ ] : char, author[ ] : char, publisher[ ] : char, quantity : unsigned int, wholeSale : double, retail : double, date : time_t) :
+ getISBN() : char*
+ getTitle() : char*
+ getAuthor() : char*
+ getPublisher() : char*
+ getQuantity() : unsigned int
+ getWholeSale() : double
+ getRetail() : double
+ getDate() : time_t
+ print() : void
+~Book():
Inventory
+ inventoryMenu() : void
+ sortByTitle(booklist : vector<Book>) : void
+ sortByQuantity(booklist : vector<Book>) : void
+ sortByCost(booklist : vector<Book>) : void
+ sortByAge(booklist : vector<Book>) : void
+ lookUpBookISBN(isbn[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookTitle(title[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookAuthor(author[ ] : char, booklist : vector<Book>) : vector<int >
+ lookUpBookPublisher(publisher[ ] : char, booklist : vector<Book>) : vector<int >
+ addBook(booklist : vector<Book>) : void
template(class T1&, T2)
+ setValue(dest : T1, source : T2) : void
+ editBook(location : int, booklist : vector<Book>) : void
+ readList() : vector<Book>
+ writeList(list : vector<Book>) : void
+ ~Inventory() :
Reports
+ menu() : void
+ printList(booklist : vector<Book>) : void
Reports Menu Screen
Inventory Menu Screen

Inheritance:



Pseudocode:

```
==============================================================================
                                   Main.cpp
==============================================================================
Include iostream
Include fstream
Include string
Include ctime
Include all .h files for other classes/modules
using namespace std
int main
    call splashscreen function
    create an inventory, cashier, and report object to call their menu functions
    create pointers for inventory, cashier, and report
    create int variable "choice" and initialize it to 0
    open booklist for both input and output
    while user doesn't select option 4 to exit program
        clear screen
        create time struct rawtime and pointer to it called timeinfo using ctime
        create char array to hold timeinfo data, format date and time
        display date and time
        use cout to display main menu, options are cashier, inventory, report, and exit
        cin to get choice value
        if user doesn’t enter an int value
            clear cin buffer
            set choice variable to 0 again
            cin.ignore to ignore newlines without input
        while choice value is less than 1 or greater than 4
            display message telling user to enter 1-4
            get choice value again
        switch statement for int choice
        case 1:
            open cashier menu using pointer to cashier object
            break;
        case 2:
            open inventory menu using pointer to inventory object
            break;
        case 3:
            open report menu using pointer to report object
            break;
        case 4:
            break;
    display exit message for when user enters 4 and main menu ends loop
    system pause so user can see the exit message
    return 0 to exit program
splashscreen
    welcome message and ascii art
    system pause so user can bask in ascii glory
```

```
================================================================
                            Book.cpp
================================================================
```

```
Book constructor
    strcpy to initialize sISBN with char[] “isbn” parameter
    strcpy to initialize sTitle with char[] “title” parameter
    strcpy to initialize sAuthor with char []“author” parameter
    strcpy to initialize sPublisher with char[] “publisher” parameter
    sRetail = double “retail”  parameter
    sQuantity = unsigned int “quantity” parameter
    sWholesale = double “wholesale” parameter
    sDate = time_t “date” parameter
sISBN accessor function
sTitle accessor
sAuthor accessor
sPublisher accessor
sQuantiy accessor
sWholesale accessor
sRetail accessor
sDate accessor
Book print function
    cout each private member of book class
    printf wholesale and retail price formatted with $ and 2 decimal places
    display date added as time struct
    has a boolean argument to indicate if cashier menu is using the print
        if true, do not display wholesale price and date added
Book destructor

======================================================================
                              Menu.cpp
======================================================================
include menu.h
virtual function menu() used as parent to allow for polymorphism

======================================================================
                            Inventory.cpp
======================================================================
include Inventory.h, Book.h, vector, and fstream
using namespace std
inventory menu
    int inventoryChoice initialized to 0
    do while loop to get which Inventory function to execute
        clear screen
        display date and time
        call readList function to get booklist
        cout menu options: look up, add, edit, and delete book, exit
        cin inventoryChoice value
        clear cin if non number is entered
        switch statement for inventoryChoice
        case1:
                another int choice
                another do while loop to get input
                clear screen
                display date and time
                Look Up Book menu: user chooses which field to look up a book by:
                ISBN, Author, Publisher, Title, Return to Previous Menu
                cin choice
                switch statement for this choice
                case1:
                    try to search isbn user inputs using lookUpBook
                    catch error if no books found
                    system pause
                case2:
                    try to search title user inputs using lookUpBook
                    catch error if no books found
                    system pause
                case3:
                    try to search author user inputs using lookUpBook
                    catch error if no books found
                    system pause
                case4:
                    try to search publisher user inputs using lookUpBook
                    catch error if no books found
                    system pause
                case 5:
                    break
            case 2:
                call addbook function
            case3:
                identical to case 1 of this switch, calls editBook instead of lookUpBook
            case4:
                identical to case 1 of this switch, calls deleteBook instead
                Also, prompts user to make sure they want to delete the book before delete
            case5:
                break - returns user to main menu
            default: asks user to enter 1-5
                system pause
        loops while choice isn’t 5
Inventory sortByTitle
```

```
        loops through booklist according to its size, using counter i
                loops through booklist according to its size again, using counter j
                        if booklist at i’s title is greater than booklist at j’s title
                                creates 2 temp book objects for book at i and j
                                erases book at position j
                                inserts book at i into j’s position
                                inserts book at position j into position i
                                removes book after newly inserted book
Inventory sortByQuantity
        exact same but uses getQuantity instead of getTitle Book function, to sort by quantity
Inventory sortByCost
        exact same but uses getRetail instead of getTitle Book function, to sort by cost
Inventory sortByAge
        exact same but uses getDate instead of getTitle Book function, to sort by age/date
Inventory lookUpBookISBN
        takes in a boolean argument that indicates if cashier menu is using lookUp
        cout “Search Results”, end line
        declare vector of ints called foundBooks
        declare int count, set to 0
        loop while counter i < size of booklist
                makes all strings lowercase
                compares isbn given to every Book’s sISBN in booklist file
                if a match is found,
                        adds that match’s location as an int to the int vector foundBooks
                        prints the current number of found books to the screen
                        prints the current book to the screen next
                                if cashier boolean is true, passes bool to Book print() function
                        endline

        If no books were found, throws an exception back to the inventory menu function
        returns the foundBooks int vector
Inventory lookUpBookTitle
        exact same but uses getTitle instead of getISBN public Book function to look up by title
Inventory lookUpBookAuthor
        exact same but uses getAuthor instead of getISBN to look up by author
Inventory lookUpBookPublisher
        exact same but uses getPublisher instead of getISBN to look up by publisher
Inventory addBook
        takes a vector of books called booklist as sole parameter
```

```
    clear screen
    initialize validChoice boolean variable as false
    declare char arrays for isbn, title, author, and publisher
    declare int variable for quantity, double for wholesalecost, and double for retailcost
    get date using ctime
    display date
    display addBook function's menu, asking for all of book's private member vars 1 by 1
    user can enter 0 to exit this process at any time
    uses cin getline function and cin ignore function to ensure that input is correct
    only gets everything after isbn if there isbn is not equal to 0
        program will check if user attempts to enter an ISBN already in the database
        if a match is found, request for a new ISBN or user can enter 0 to cancel adding
    uses the validChoice bool declared earlier to check input validity for quantity of book,
    while validChoice is false
        get quantity value from cin, check to see if it is a number or not
        if it's not
            clear cin and set value of quantity to -1
            cin.ignore
        if the value of quantity is less than 0, use cout to display error message
        or else set validChoice equal to true
    exact same process for getting wholesale cost and retail price as quantity
    create book object using all of the input just received from user to initialize private vars
    push_back the book into the booklist vector
    write the booklist vector to the file using writelist function
    display a confirmation message and use system pause so user can see it
Template used for setting sQuantity and sRetail within editBook function
    two classes, T1 and T2
    T1 is passed by reference so that it can be altered
    set T1's value to the value of T2
Inventory editBook function
    parameters are an int called location and a vector of books called booklist
    declare choice variables to get user's choice
    clear screen and display time
    menu prompts user by first showing the book's information using Book print function
    then asking which variable of the book the user would like to edit
    user can edit all but date added and wholesale cost
    gets user's choice and makes sure that it is an int, defaults to 0 if non-number entered
    based on user's choice, switch statement prompts user for the new isbn/title/author/etc
    for vars stored in char arrays, simply uses getline and then uses strcpy to overwrite old
    for ISBN, program will check if new ISBN is already in database for another book
        does not return a match if the found ISBN is the same as the one being edited
    for quantity and retail cost, program checks input to make sure it is valid number
    after replacing old data, uses writelist function to save the altered list
    displays message confirming change and uses system pause to allow user to see it
Inventory readList
    accepts no parameters, returns vector of books
    declares int numBooks, this will be the first thing in the booklist file
    declare vector for books called booklist
    open ifstream "ifs" for booklist.txt as binary input
    if ifs has data in it
        read number of Book objects in the file (at beginning) into numBooks variable
        check to make sure memory could be allocated for the booklist.txt
        if pointer the book vector is a nullptr
            display error message, pause screen, then exit
        for loop extending length of file determined by numBooks var
            create temp book object
            read a single book object from the file into the temp book object
            push_back the book object into the booklist vector
    close ifs input file
    return the booklist vector
Inventory writeList
    no return, accepts vector of books as parameter
    open ofstream "ofs" for booklist.txt file as binary output
    declare int var numBooks equal to the booklist's size(number of books in it)
    loop numBooks times
        write each book object to the file
    close the ofs file
```

==

Cashier.cpp

==

```
include iomanip, cstring, vector, book.h, inventory.h, cashier.h
using namespace std
Cashier menu
    declare Book object book
    initialize vector of Book by calling Inventory readList function, inherited from inv class
    declare int choice var
    do while loop to loop the menu
```

```
        clear screen, display time, display menu
        menu options: add book to cart, edit the cart, checkout, and exit
        get choice from user and check to make sure its valid
        switch statement based on int choice:
        case1:
            for addBook, first ask user how they wish to search for the book
            works same as inventory menu, with its own switch statement,
            option for searching for an isbn, title, author, or publisher
            each one of these is identical, uses exception handling first to try calling
                the corresponding lookUpBook function from inventory
                if it can, displays the search results and asks user which to call
                    results omit wholesale cost and date added displays
                addToSale function with, checks that input (int) validity
                tries to add to sale, catches exception if there are 0 of that book
            catches exception book was not found by lookUpBook function
            uses system pauses after exceptions could be caught so user can read them
        case2:
            for editing cart, first checks cart size
            if it is empty, displays error and pauses
            else, clears screen and displays time and menu displaying current cart
            declares choice int
            menu asks which book to remove and user can enter 0 to cancel
            cin to get choice
            defaults choice to larger than array size if input isn't a non-unsigned int
            if choice is larger than array size, displays error message
            calls subFromSale function with the choice and the booklist vector
        case3:
            for checkout, first checks to ensure cart is not empty
            if cart isn't empty, calls Checkout Cashier function
            clears salelist vector which addToSale function generates
            writes the booklist vector to the file
        case4:
            doesn't exit immediately if books still in cart, asks user if they are sure
            uses 1 for yes, 0 for no
            if 1, clears salelist function and then user can exit because cart is empty
            checks user input to make sure they enter 1 or 0
        default:
            display error message asking for 1, 2, 3, or 4, then pause
Cashier addToSale
```

```
    parameters are int location, vector of books called booklist passed by reference
    declare int variables number equal to -1 and quantity equal to 0
    get quantity value by calling getQuantity book function for book at location given
    if quantity = 0, throw exception stating that there are no books left
    else
    asks for number of copies to add to cart
    while number is less than 0
        cin to get number, check input validity
        if number entered is greater than quantity of book, asks user if they wish to add all
        1 for yes, 0 for no, checks input
        if 1, sets number value equal to quantity value (add all of the book to cart)
        if 0, sets number value back to -1 and asks for number again
        if number is less than 0
            display error message b/c can't add less than 0 of a book
            cin clear and ignore
    int variable inCart initialized to -1
    loops salelist.size times (number of books in cart)
        if the book is already in the cart, program sets the location in cart to inCart var
    if inCart is less than 0 (book is NOT already in the cart)
        push_back the book into the salelist vector
        push_back "number" int var into cartQuantity int vector
        push_back location of book into bookLocation vector
    else
        increase cartQuantity vector's value at location of the book in the cart by number
    reduce sQuantity variable of booklist vector's Book object we are adding by number
    display confirmation that the book was added to the cart in the right amount
Cashier subFromSale
    accepts int location and vector of Book booklist parameters
    declare number and quantity variables, number = -1 and quantity = quant to be sold
    asks user how many they wish to remove
    get number from user with cin, check that it is a number
    if number is negative, asks for a positive number
    if number is greater than the number of books in the cart, sets the number = to quantity
    adds back the number of books to inventory list from cart
    reduces the number of books in cart by number
    display message confirming removal
    if the quantity of the book in the cart is now 0
        erase the book object from salelist vector
        erase from bookLocation and cartQuantity int vectors as well
```

```
Cashier Checkout
    initialize double total = to 0
    clear screen and display time, display title and column information
    for the number of books in the salelist vector
        add the cost of the book times the quantity of that book to the total var
        display the book's information, including total cost of sale (book cost*quant)
    display subtotal, current value of total
    display tax value (const taxrate private member * total)
    display total + the tax value as final total
    system pause
```

==

Reports.cpp

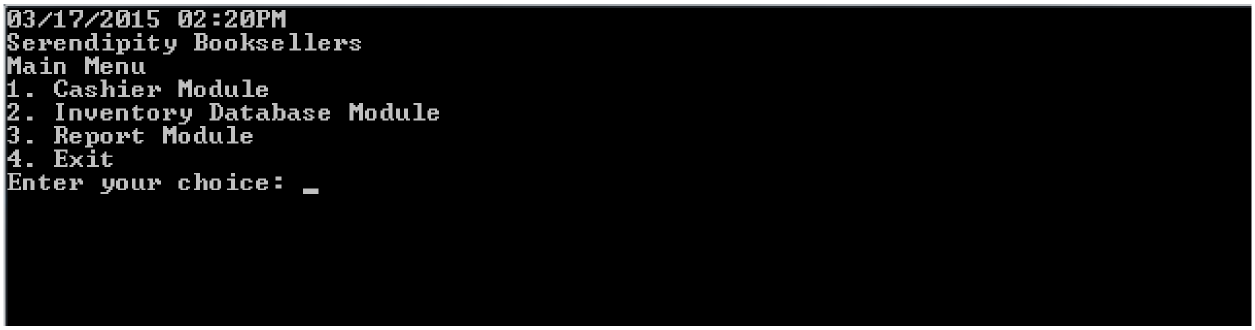
==

```
include reports.h, iostream, fstream, iomanip
using namespace std
Report menu
    set int choice = 0
    while choice is not equal to 7
        clear screen, display time and date, display menu options:
        Inventory listing, inventory wholesale listing, inventory retail value
        listing by quantity, listing by cost, listing by age
        get choice
        reads booklist file into vector booklist
        calls sortByTitle function with booklist vector
        clear screen
        switch statement for choice
        case1:
            display time and formatting, then call printlist function
            system pause
            break
        case 2:
            display time and screen formatting
            declare double total
            use loop to print each book's information
                calculate total wholesale cost of book, set double cost to this value
                temporary string holds title with getTitle then truncates title if long
                display book's name, quantity, wholesale cost, and double cost
                add the total wholesale cost of book to total variable
            prints entire inventory's wholesale cost
```

```
                system pause
                break
        case3:
                identical to previous but uses retailCost variable instead of wholesaleCost
        case4:
                lists by quantity, first displaying date/time and screen formatting with title
                call sortByQuantity function with booklist vector
                call printList function with booklist vector
                system pause
                break
        case5:
                display date/time and titles
                lists by cost:
                calls sortByCost function
                calls printList function
                system pause
                break
        case6:
                display date/time and titles, formatting
                lists by age:
                calls sortByAge function
                calls printList function
                system pause
                break
        case 7:
                break
        default:
                error message asks user to enter a number 1-7
                system pause
Report printList
        accepts vector of Book booklist as parameter, returns nothing
        if booklist's size is 0
                tell user that there are no books in the inventory
                format with line below message
        else
                for each book in the inventory
                        call print Book function
                        print a line of hyphens
```

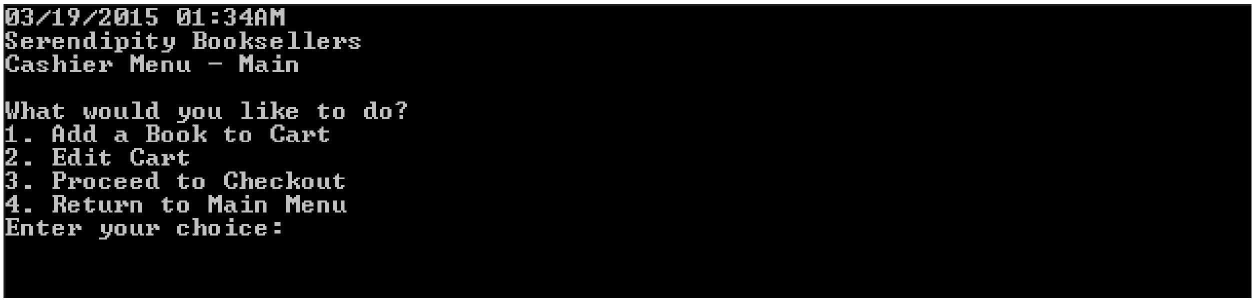
Screenshots:

Main Menu:



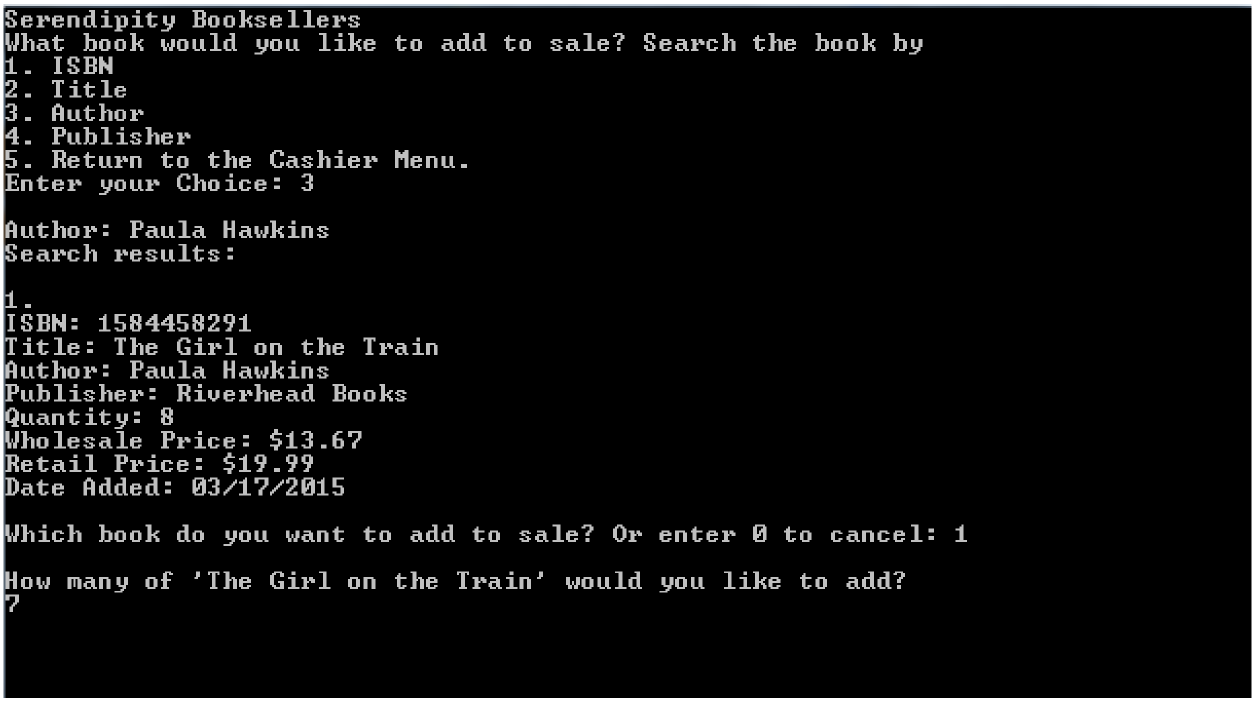
```
03/17/2015 02:20PM
Serendipity Booksellers
Main Menu
1. Cashier Module
2. Inventory Database Module
3. Report Module
4. Exit
Enter your choice: _
```

Cashier:



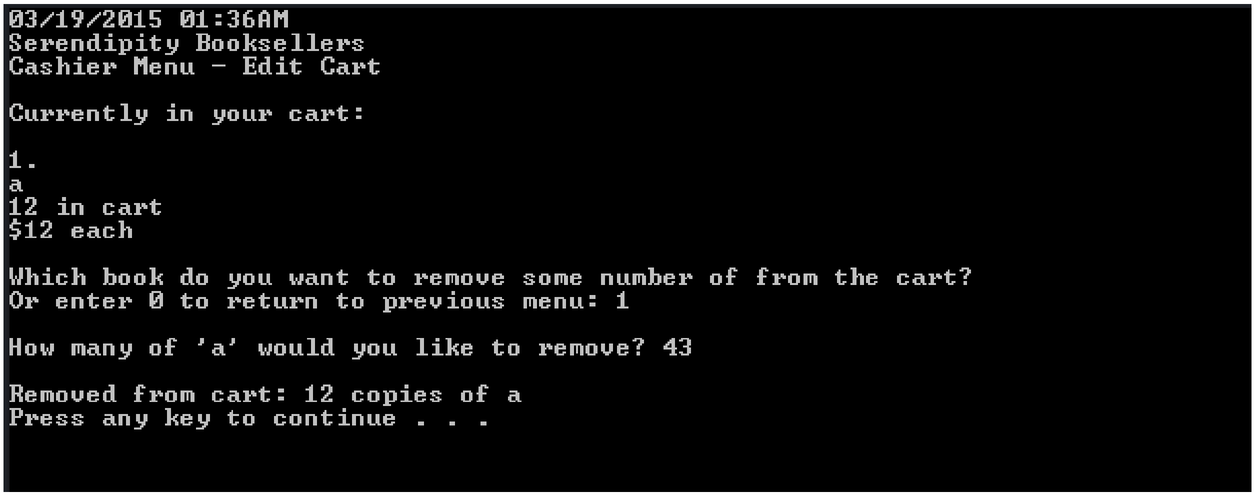
```
03/19/2015 01:34AM
Serendipity Booksellers
Cashier Menu - Main

What would you like to do?
1. Add a Book to Cart
2. Edit Cart
3. Proceed to Checkout
4. Return to Main Menu
Enter your choice:
```

Add book to Cart:



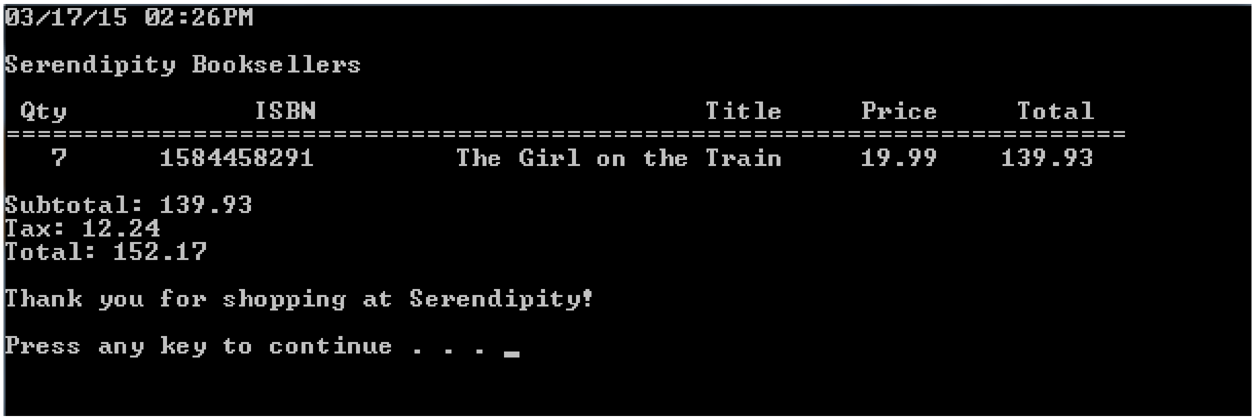
```
Serendipity Booksellers
What book would you like to add to sale? Search the book by
1. ISBN
2. Title
3. Author
4. Publisher
5. Return to the Cashier Menu.
Enter your Choice: 3

Author: Paula Hawkins
Search results:

1.
ISBN: 1584458291
Title: The Girl on the Train
Author: Paula Hawkins
Publisher: Riverhead Books
Quantity: 8
Wholesale Price: $13.67
Retail Price: $19.99
Date Added: 03/17/2015

Which book do you want to add to sale? Or enter 0 to cancel: 1

How many of 'The Girl on the Train' would you like to add?
7
```

Edit cart:



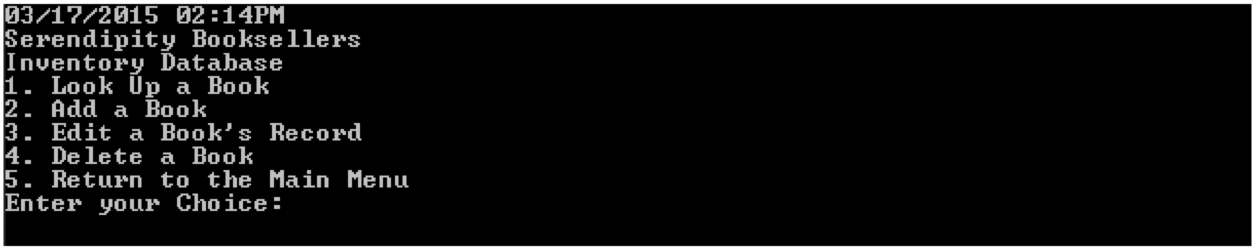
```
03/19/2015 01:36AM
Serendipity Booksellers
Cashier Menu - Edit Cart

Currently in your cart:

1.
a
12 in cart
$12 each

Which book do you want to remove some number of from the cart?
Or enter 0 to return to previous menu: 1

How many of 'a' would you like to remove? 43

Removed from cart: 12 copies of a
Press any key to continue . . .
```

Checkout:



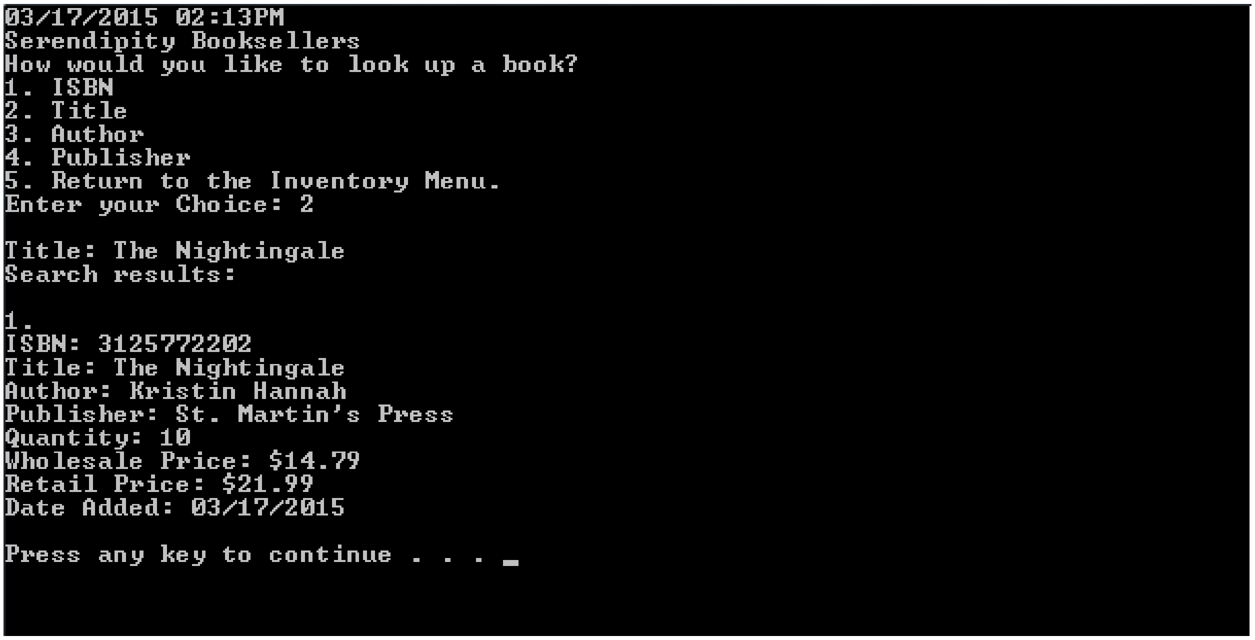
```
03/17/15 02:26PM

Serendipity Booksellers

 Qty            ISBN                         Title     Price     Total
==================================================================
   7      1584458291         The Girl on the Train      19.99    139.93

Subtotal: 139.93
Tax: 12.24
Total: 152.17

Thank you for shopping at Serendipity!

Press any key to continue . . .
```

Inventory Menu:



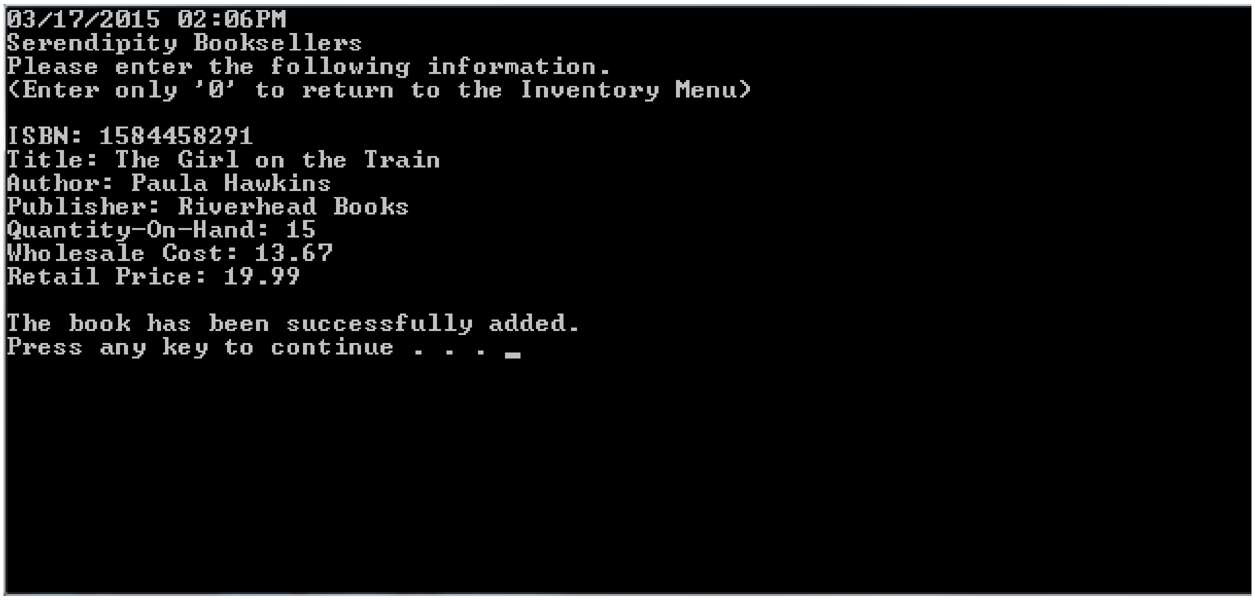
```
03/17/2015 02:14PM
Serendipity Booksellers
Inventory Database
1. Look Up a Book
2. Add a Book
3. Edit a Book's Record
4. Delete a Book
5. Return to the Main Menu
Enter your Choice:
```

Look Up Book:



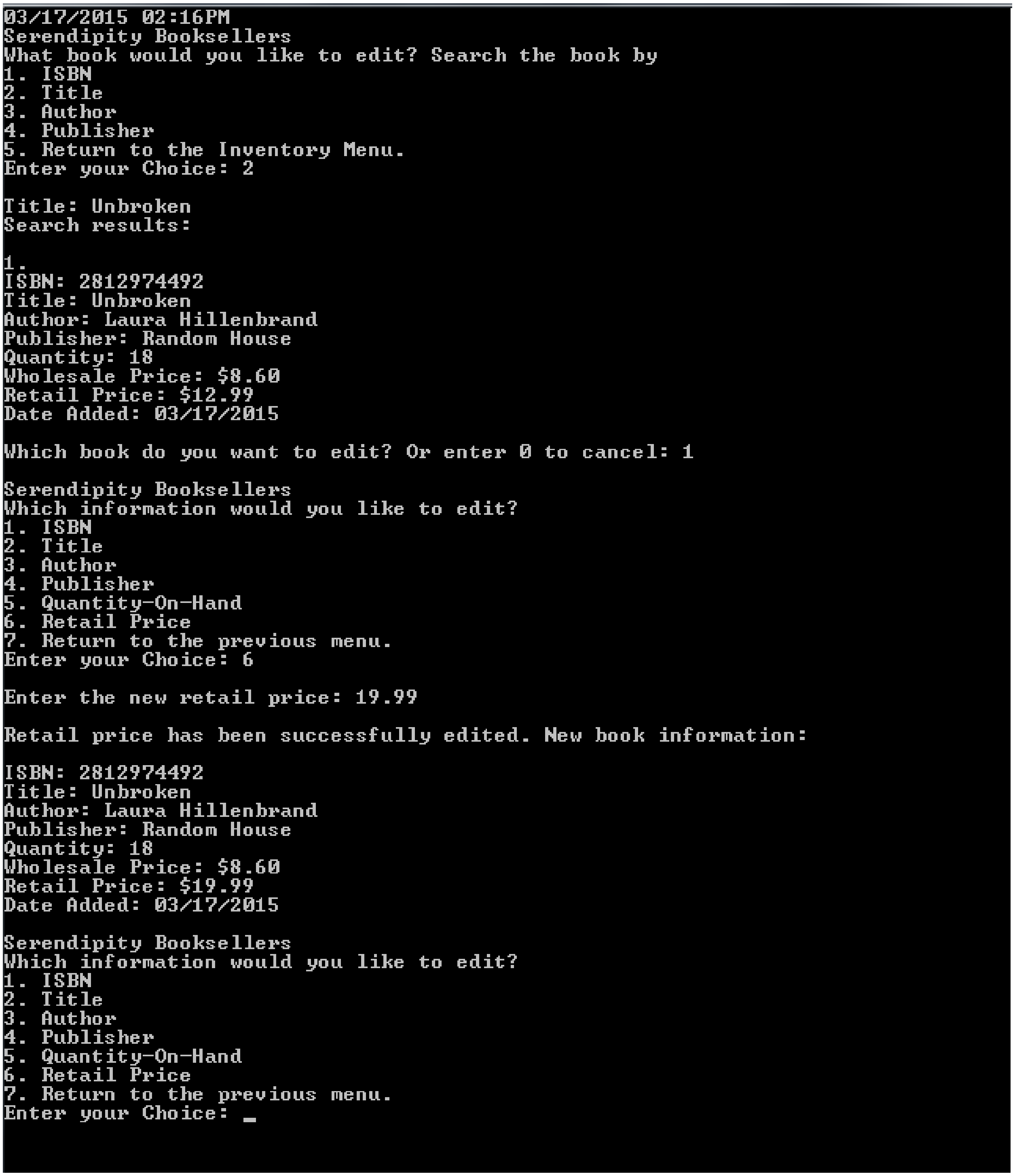
```
03/17/2015 02:13PM
Serendipity Booksellers
How would you like to look up a book?
1. ISBN
2. Title
3. Author
4. Publisher
5. Return to the Inventory Menu.
Enter your Choice: 2

Title: The Nightingale
Search results:

1.
ISBN: 3125772202
Title: The Nightingale
Author: Kristin Hannah
Publisher: St. Martin's Press
Quantity: 10
Wholesale Price: $14.79
Retail Price: $21.99
Date Added: 03/17/2015

Press any key to continue . . .
```

Add Book:



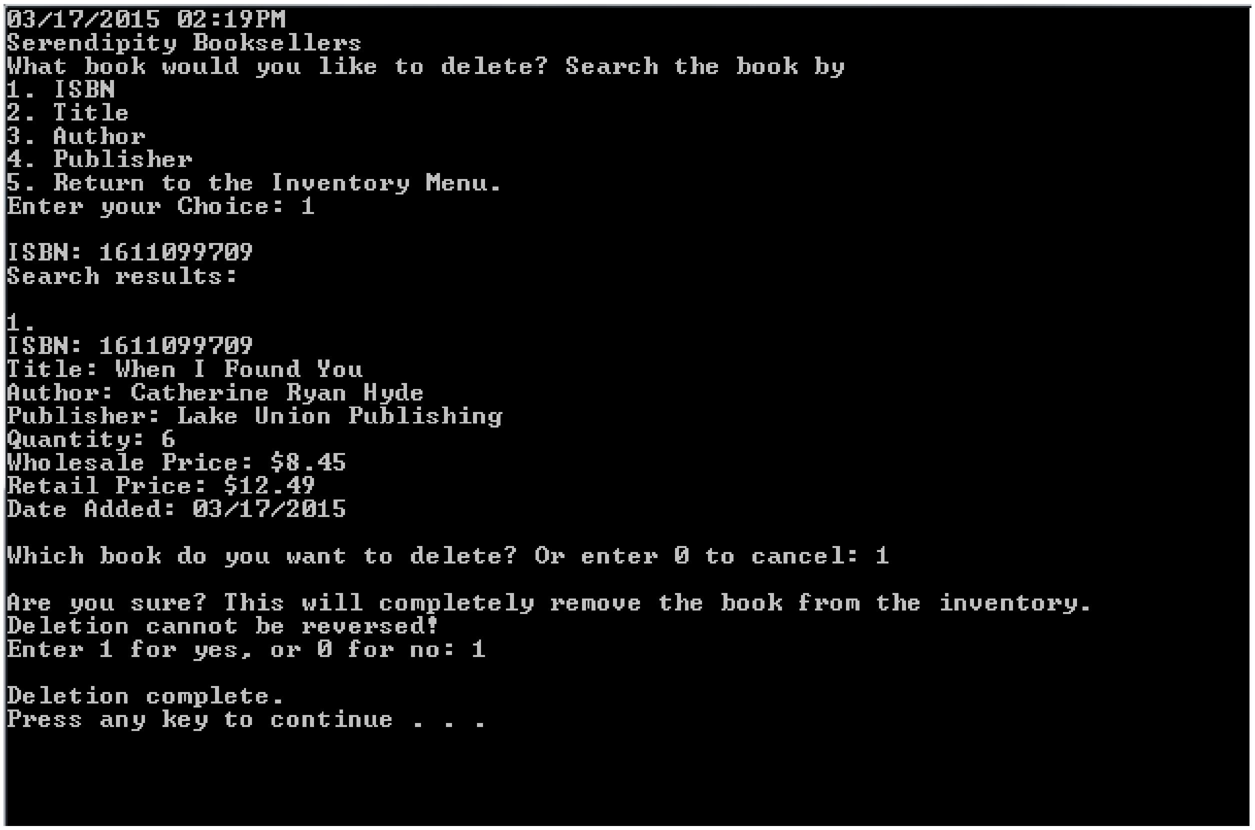
```
03/17/2015 02:06PM
Serendipity Booksellers
Please enter the following information.
(Enter only '0' to return to the Inventory Menu)

ISBN: 1584458291
Title: The Girl on the Train
Author: Paula Hawkins
Publisher: Riverhead Books
Quantity-On-Hand: 15
Wholesale Cost: 13.67
Retail Price: 19.99

The book has been successfully added.
Press any key to continue . . .
```

Editing a Book:



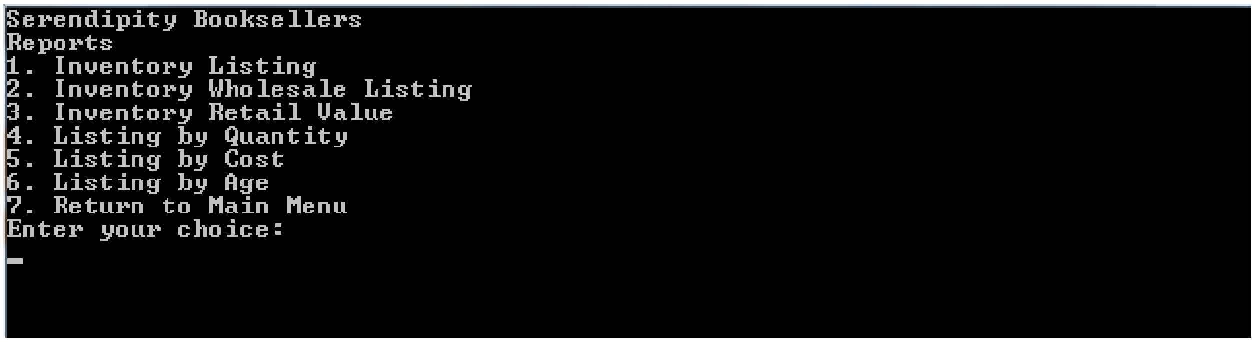
```
03/17/2015 02:16PM
Serendipity Booksellers
What book would you like to edit? Search the book by
1. ISBN
2. Title
3. Author
4. Publisher
5. Return to the Inventory Menu.
Enter your Choice: 2

Title: Unbroken
Search results:

1.
ISBN: 2812974492
Title: Unbroken
Author: Laura Hillenbrand
Publisher: Random House
Quantity: 18
Wholesale Price: $8.60
Retail Price: $12.99
Date Added: 03/17/2015

Which book do you want to edit? Or enter 0 to cancel: 1

Serendipity Booksellers
Which information would you like to edit?
1. ISBN
2. Title
3. Author
4. Publisher
5. Quantity-On-Hand
6. Retail Price
7. Return to the previous menu.
Enter your Choice: 6

Enter the new retail price: 19.99

Retail price has been successfully edited. New book information:

ISBN: 2812974492
Title: Unbroken
Author: Laura Hillenbrand
Publisher: Random House
Quantity: 18
Wholesale Price: $8.60
Retail Price: $19.99
Date Added: 03/17/2015

Serendipity Booksellers
Which information would you like to edit?
1. ISBN
2. Title
3. Author
4. Publisher
5. Quantity-On-Hand
6. Retail Price
7. Return to the previous menu.
Enter your Choice: _
```

Deleting a Book:



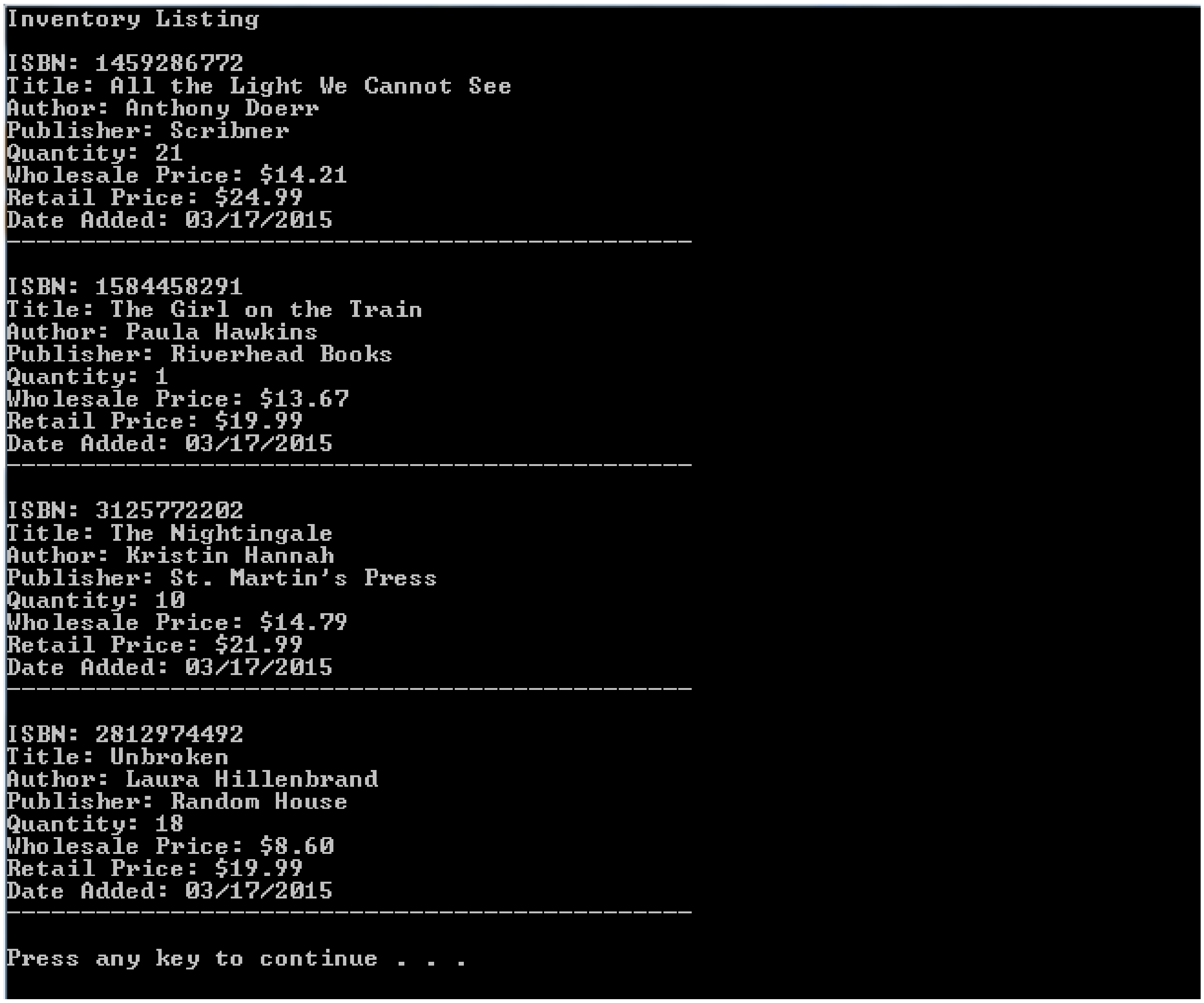
```
03/17/2015 02:19PM
Serendipity Booksellers
What book would you like to delete? Search the book by
1. ISBN
2. Title
3. Author
4. Publisher
5. Return to the Inventory Menu.
Enter your Choice: 1

ISBN: 1611099709
Search results:

1.
ISBN: 1611099709
Title: When I Found You
Author: Catherine Ryan Hyde
Publisher: Lake Union Publishing
Quantity: 6
Wholesale Price: $8.45
Retail Price: $12.49
Date Added: 03/17/2015

Which book do you want to delete? Or enter 0 to cancel: 1

Are you sure? This will completely remove the book from the inventory.
Deletion cannot be reversed!
Enter 1 for yes, or 0 for no: 1

Deletion complete.
Press any key to continue . . .
```

Reports Menu:



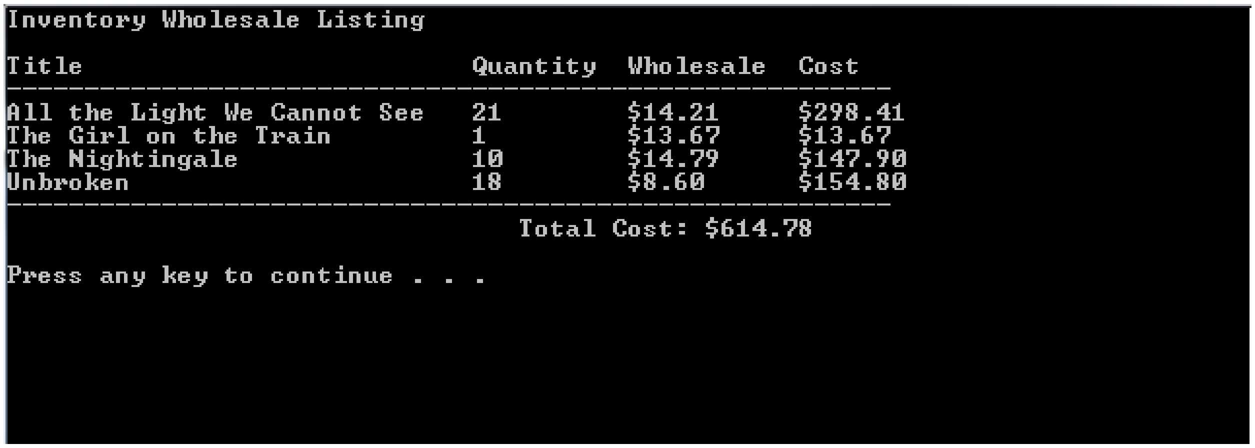
```
Serendipity Booksellers
Reports
1. Inventory Listing
2. Inventory Wholesale Listing
3. Inventory Retail Value
4. Listing by Quantity
5. Listing by Cost
6. Listing by Age
7. Return to Main Menu
Enter your choice:
_
```

Inventory Listing:



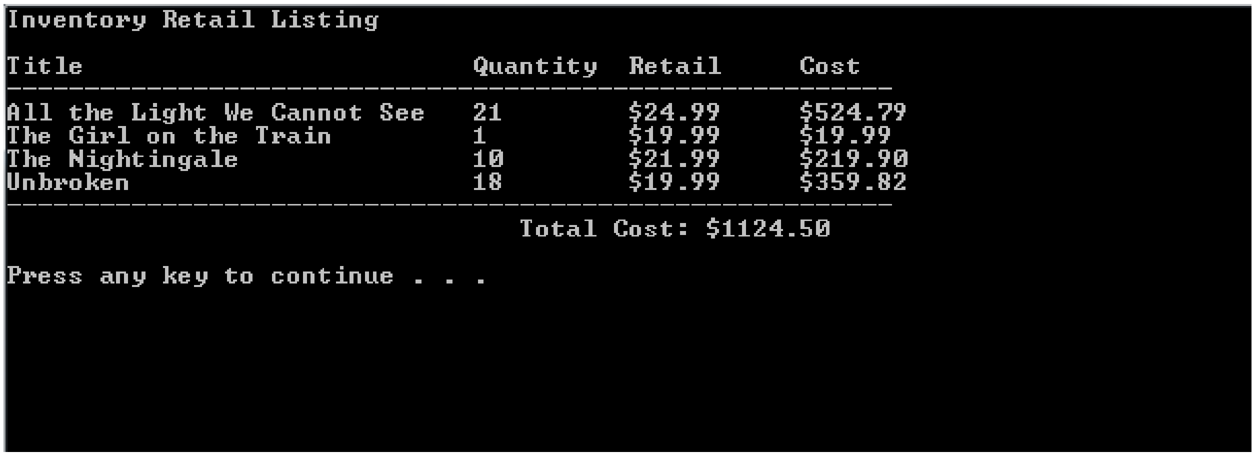
```
Inventory Listing

ISBN: 1459286772
Title: All the Light We Cannot See
Author: Anthony Doerr
Publisher: Scribner
Quantity: 21
Wholesale Price: $14.21
Retail Price: $24.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 1584458291
Title: The Girl on the Train
Author: Paula Hawkins
Publisher: Riverhead Books
Quantity: 1
Wholesale Price: $13.67
Retail Price: $19.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 3125772202
Title: The Nightingale
Author: Kristin Hannah
Publisher: St. Martin's Press
Quantity: 10
Wholesale Price: $14.79
Retail Price: $21.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 2812974492
Title: Unbroken
Author: Laura Hillenbrand
Publisher: Random House
Quantity: 18
Wholesale Price: $8.60
Retail Price: $19.99
Date Added: 03/17/2015
----------------------------------------------

Press any key to continue . . .
```

Inventory Wholesale Listing:



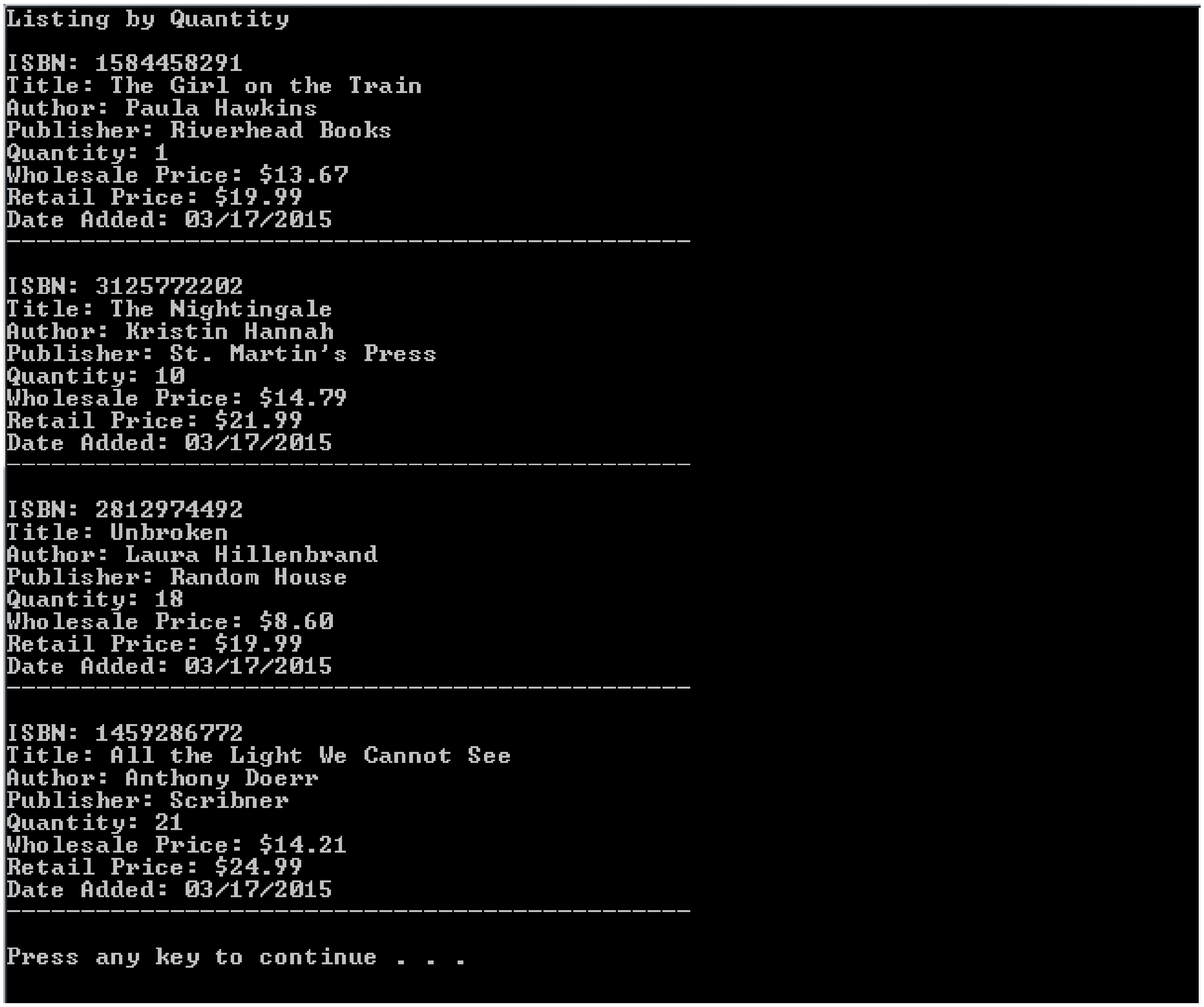
```
Inventory Wholesale Listing

Title                         Quantity  Wholesale  Cost
---------------------------------------------------------
All the Light We Cannot See   21        $14.21     $298.41
The Girl on the Train         1         $13.67     $13.67
The Nightingale               10        $14.79     $147.90
Unbroken                      18        $8.60      $154.80
---------------------------------------------------------
                                 Total Cost: $614.78

Press any key to continue . . .
```

Inventory Retail Value:



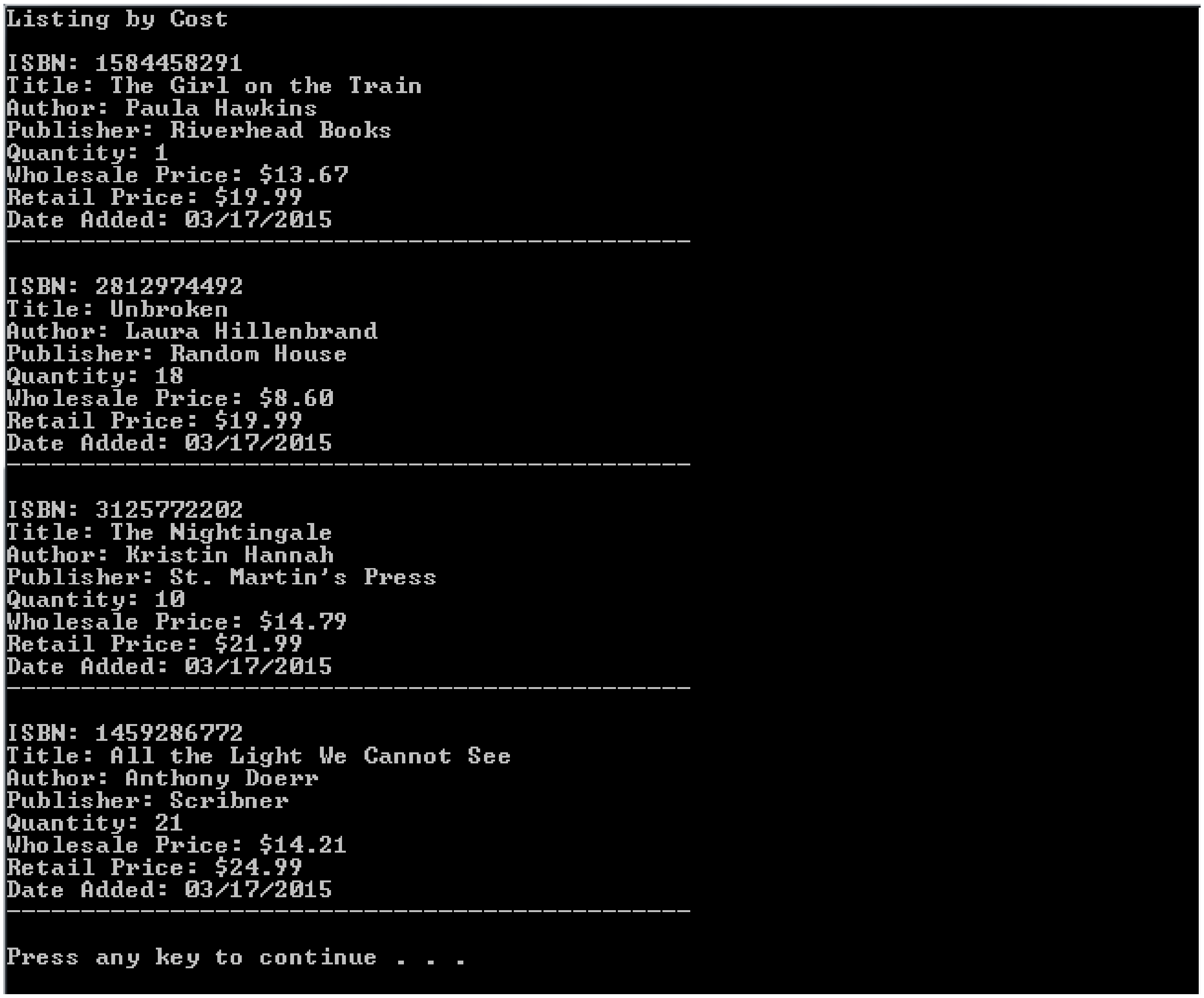
```
Inventory Retail Listing

Title                         Quantity  Retail     Cost
---------------------------------------------------------
All the Light We Cannot See   21        $24.99     $524.79
The Girl on the Train         1         $19.99     $19.99
The Nightingale               10        $21.99     $219.90
Unbroken                      18        $19.99     $359.82
---------------------------------------------------------
                                 Total Cost: $1124.50

Press any key to continue . . .
```

Listing by Quantity:



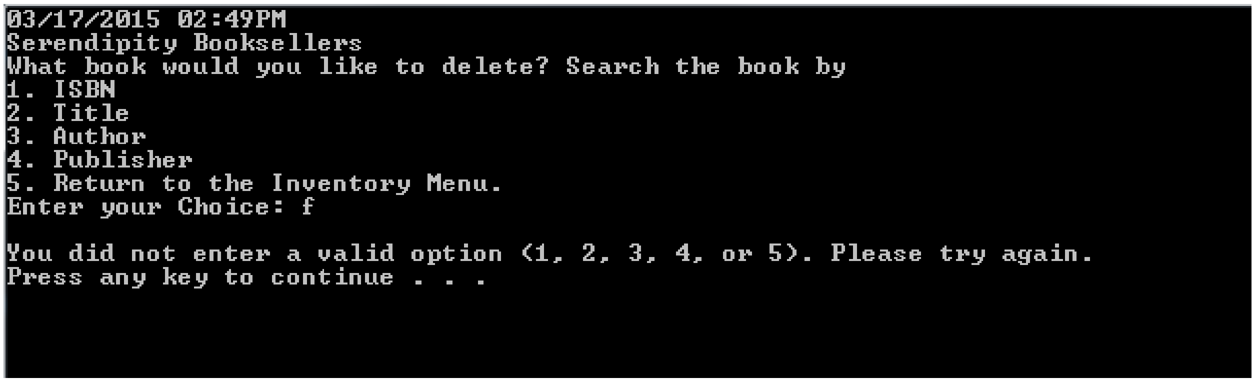
```
Listing by Quantity

ISBN: 1584458291
Title: The Girl on the Train
Author: Paula Hawkins
Publisher: Riverhead Books
Quantity: 1
Wholesale Price: $13.67
Retail Price: $19.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 3125772202
Title: The Nightingale
Author: Kristin Hannah
Publisher: St. Martin's Press
Quantity: 10
Wholesale Price: $14.79
Retail Price: $21.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 2812974492
Title: Unbroken
Author: Laura Hillenbrand
Publisher: Random House
Quantity: 18
Wholesale Price: $8.60
Retail Price: $19.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 1459286772
Title: All the Light We Cannot See
Author: Anthony Doerr
Publisher: Scribner
Quantity: 21
Wholesale Price: $14.21
Retail Price: $24.99
Date Added: 03/17/2015
----------------------------------------------

Press any key to continue . . .
```

Listing by Cost:



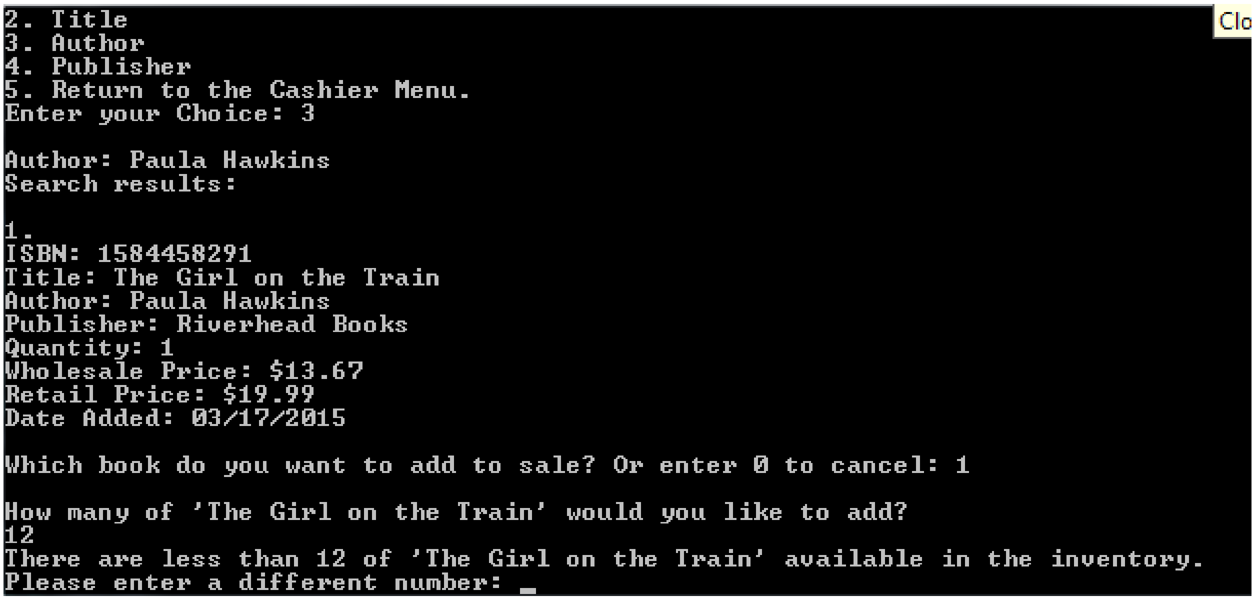
```
Listing by Cost

ISBN: 1584458291
Title: The Girl on the Train
Author: Paula Hawkins
Publisher: Riverhead Books
Quantity: 1
Wholesale Price: $13.67
Retail Price: $19.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 2812974492
Title: Unbroken
Author: Laura Hillenbrand
Publisher: Random House
Quantity: 18
Wholesale Price: $8.60
Retail Price: $19.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 3125772202
Title: The Nightingale
Author: Kristin Hannah
Publisher: St. Martin's Press
Quantity: 10
Wholesale Price: $14.79
Retail Price: $21.99
Date Added: 03/17/2015
----------------------------------------------

ISBN: 1459286772
Title: All the Light We Cannot See
Author: Anthony Doerr
Publisher: Scribner
Quantity: 21
Wholesale Price: $14.21
Retail Price: $24.99
Date Added: 03/17/2015
----------------------------------------------

Press any key to continue . . .
```

Error:



```
03/17/2015 02:49PM
Serendipity Booksellers
What book would you like to delete? Search the book by
1. ISBN
2. Title
3. Author
4. Publisher
5. Return to the Inventory Menu.
Enter your Choice: f

You did not enter a valid option (1, 2, 3, 4, or 5). Please try again.
Press any key to continue . . .
```

Error 2:



```
2. Title
3. Author
4. Publisher
5. Return to the Cashier Menu.
Enter your Choice: 3

Author: Paula Hawkins
Search results:

1.
ISBN: 1584458291
Title: The Girl on the Train
Author: Paula Hawkins
Publisher: Riverhead Books
Quantity: 1
Wholesale Price: $13.67
Retail Price: $19.99
Date Added: 03/17/2015

Which book do you want to add to sale? Or enter 0 to cancel: 1

How many of 'The Girl on the Train' would you like to add?
12
There are less than 12 of 'The Girl on the Train' available in the inventory.
Please enter a different number: _
```