Team Memebers: Matan Broner, Tommy McCormick, Jen Liu, Timo Bitterli

CIS 22B — Final Project
Spring Quarter 2018

Objective:

Our team of four was tasked with designing a working point of sales (POS) system for a hypothetical bookstore, Serendipity Booksellers. We were tasked with using concepts and techniques learned in CIS 22B, mainly being those relating to Object Oriented Programing — (OOP). The POS was to be able to perform the following baseline tasks as part fo the requirements:

1. Calculate total of sale including sales tax
2. When a book is purchased, remove it from the inventory file
3. Add, change, delete, and look up books in inventory file
4. Display various sales reports

Additionally, the program was to be split into a minimum of three modules:

1. Cashier Module
2. Inventory - Database Module
3. Report Module

We decided to go beyond the requirements and design three additional modules:

1. Book - Item Module
2. Cart Module
3. [Base] Module

Finally, the C++ concepts that were required to be implemented in the project were as follows:

1. Main OOP concepts (classes, inheritance, polymorphism)
2. Friends
3. Templates
4. Operator Overloading
5. Exception Handling

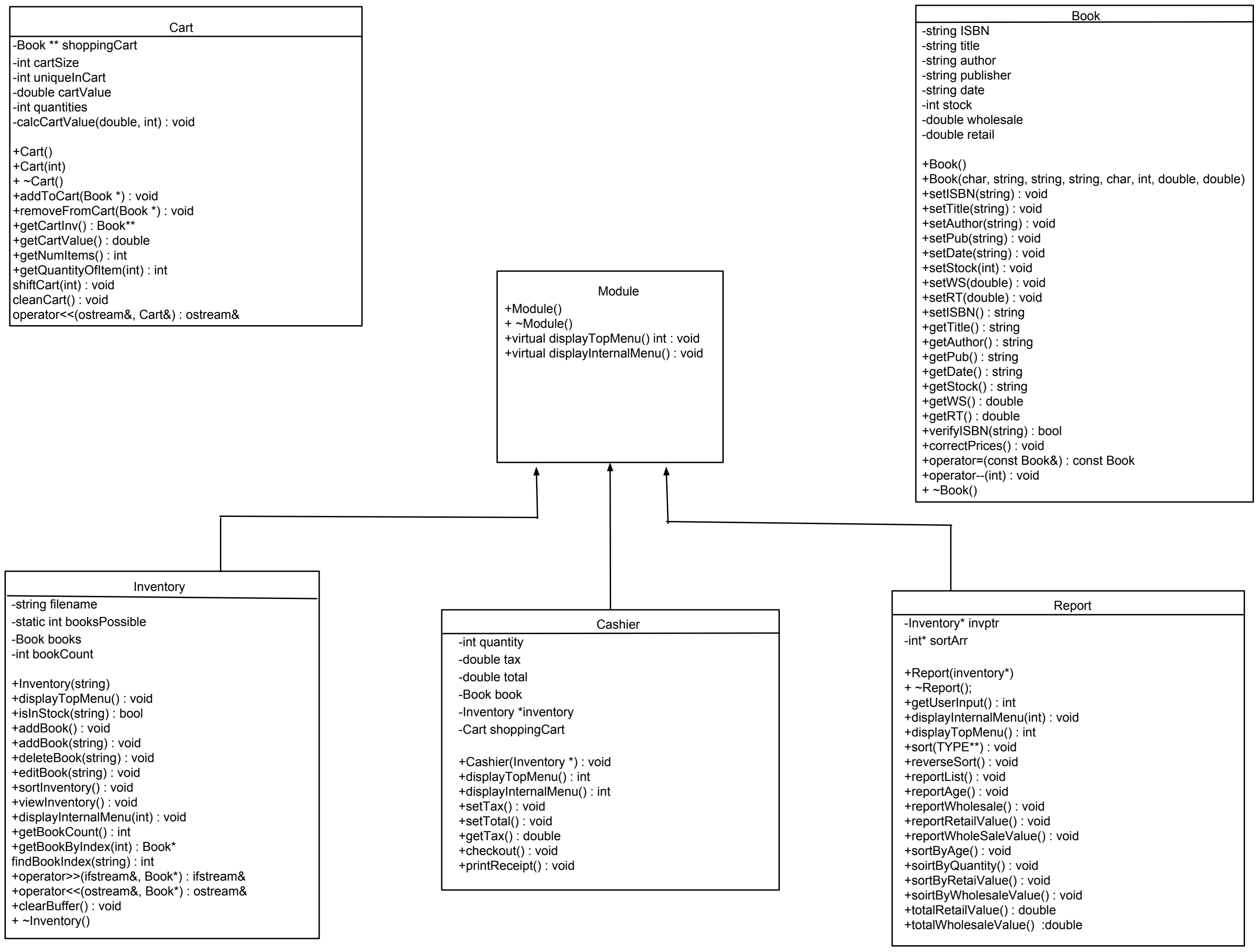
The specifications of implementing these concepts will be explained on the following page of this document.

Project Specifications — Utilization of C++ Concepts

1. Main OOP Concepts:
   A. Classes: Our entire program and its modules is built on a Module class, the others being Inventory, Cashier, Report, and Cart. Each class demonstrates the use of private member variables, creation of public getter and setter functions, and member functions meant to manipulate class behavior in various ways. We strived to create the least amount of generalized UI in the main file in our project, as we wanted each class to demonstrate its capabilities without depending on continual referencing back to the main.
   B. Inheritance: We designed our program in a fashion that the three retired modules for the project: Cashier, Inventory, and Report, all inherit from the base class Module, using the 'public' access specifier. The base Module class is meant to be quite simple, and does not actually provide any private members to the derived classes. It was useful for us in the sense that this inheritance directly allowed us to have true polymorphism, which will be explained in the next section.
   C. Polymorphism: This section is where we pulled most of our focus in terms of project design. Our base Mode class has two virtual public member functions: displayTopMenu and displayInternalMenu. Each of the three derived classes has its own version of this menu, and so it allows us to create base Module pointers in the main that are assigned to new objects of the derived classes, and run their version of the displayTopMenu function. All of the UI is done through these functions, and we truly wanted to display the usage of true polymorphism, with the notion that any classes to be written into this program must only have these two member functions, and can be easily implemented with minimal further work.
2. Friends: We wanted to be sure that we preserve the OOP concepts we used here when using friend functions. As such, we felt that the only place where direct access to a class's private members by another class would be our Report Module accessing our Inventory Module's contents. Report needs access to the original Book object pointers, and should be able to manipulate them without having to use solely getters and setters. As such, our Inventory Module lists Report as a friend class in its public member field.
3. Templates: We wanted to use templates in an area of the code where the same tsk is being repeated numerous times. The Report Module continuously performs an identical sorting algorithm using different parameters to sort. As such, we implemented a sort template function in the Report Module, which is then used to perform the various search functions (by date, by age, by price, etc.) without needing to design various member functions.
4. Operator Overloading: We added operator overloading in almost every class we wrote int he program. The Inventory class overloads the input (>>) operator to read from the data file, the Book class overloads the post increment (++) operator to add to its stock, most of the classes have an overloaded output operator (<<) to dip their contents neatly to the screen, etc. There are various examples of operator overloading, both for visually improving our code and output, and some being for truly easier to write code where it made sense.
5. Exception Handling: The Inventory Module has two prominent examples of exception handling. The first is when a user attempts to delete a book that isn't in the Inventory, the string message is thrown to the screen that this is the case. Secondly, if a user attempts to get information for a book that isn't in the inventory, the program will tell them, and even offer them to add this title to their inventory. We did this using an overloaded addBook function in the inventory that takes in this searched title.

The following pages are the pseudocode for the program, its modules, and their UML's respectively.

Main Project Pseudocode and UML’s



UML displays the inheritance of three required classes from base Module class, and the standalone item classes which serve as data storage for the required classes and their required functionality.
Use of polymorphism plays a vital role here, as you can observe the shared displayTopMenu and displayInternalMenu functions that the four related classes share.

Pseudocode: Create four base class pointers, create instance of each of the related objects using pointers, run displayTopMenu using a created Inventory object linked to one of the Module pointers.

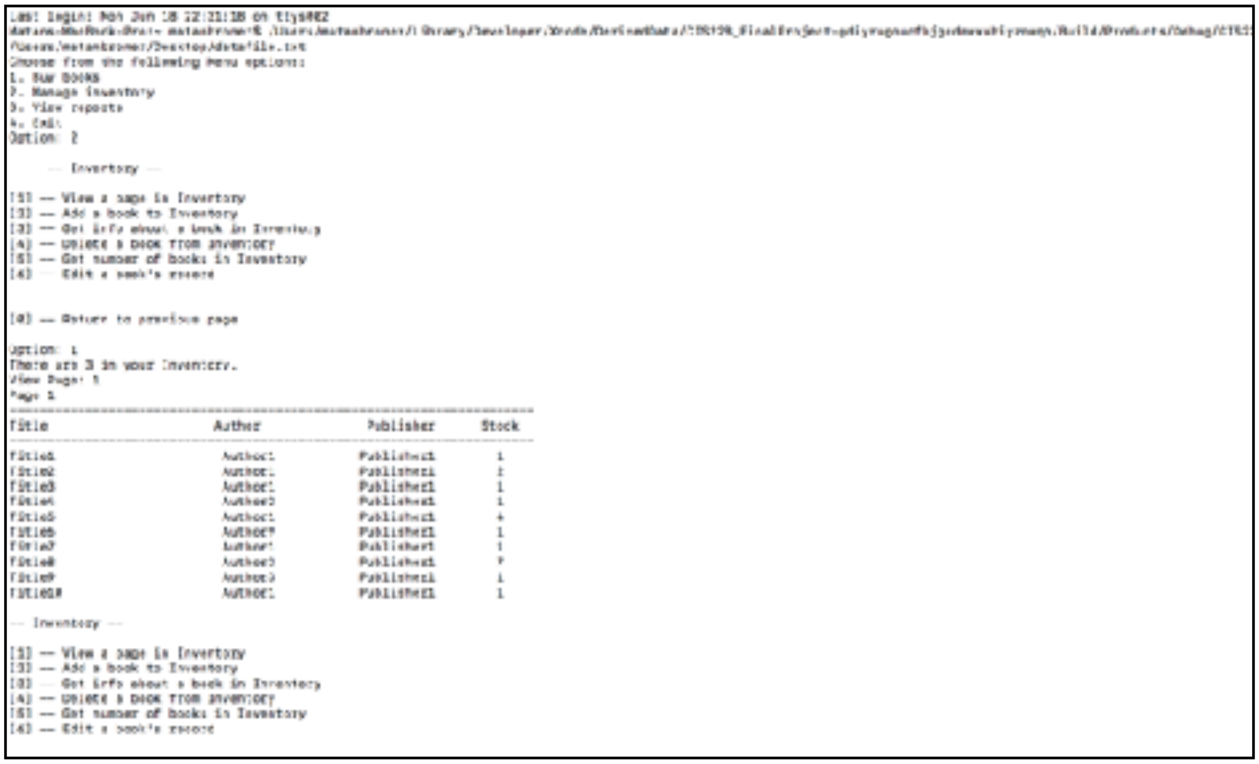
Inventory
UML and Screenshots

Inventory Module

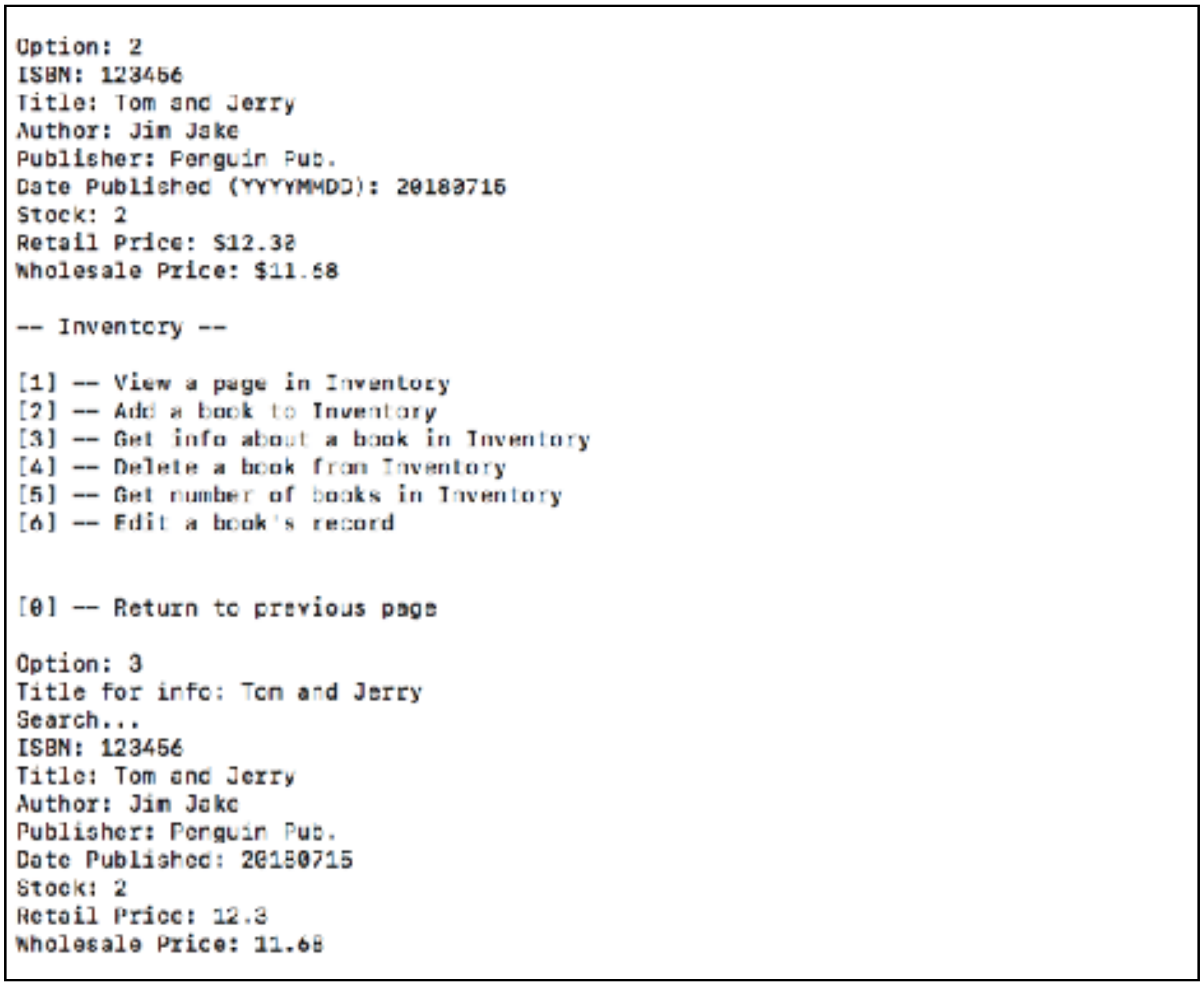
-string filename // input file name for Book object creation
-books[ ] // all books and their information read from input file
-booksPossible // number of books we allow to be inputed (ex. 100) to prevent overflow

+Inventory(string) // constructor
+displayTopMenu( ) : int // function to accept user input
+displayInternalMenu(int) : void // function to interpret displayTopMenu output
+isInStock(string title) : int // checks a Book's stock and returns how many are available
+printToFile( ) : void
+addBook( ) : void // add book to inventory
+addBook(string) : void // adds new book with given title
+deleteBook(string title) // deletes book based on inputed title from user
+editBook(string title) // allows user to change a book's information with an inputed title
+getBookInfo(string title) : void // search for book and if found displays info
+sortInventory( ) : void // a function to sort the inventory for easier searching by other functions
+viewInventory( ) : void // offers page by page view of current inventory
+getBookCount( ) : int // gets current number of created books
+findBookIndex(string) : int // finds the position index of a book with given title
+getBookByIndex(int) : Book* // offers the book pointer in the inventory with given title
+operator>>(ifstream&, Book*) : ifstream& // allows reading from file directly into book
+operator<<(ostream&, Book*) : ostream& // allows printing of all Book info easily
+clearBuffer( ) : void // clears cin buffer which prevents user input
+ ~Inventory( ) // destructor, deletes all dynamic Book objects created by Inventory

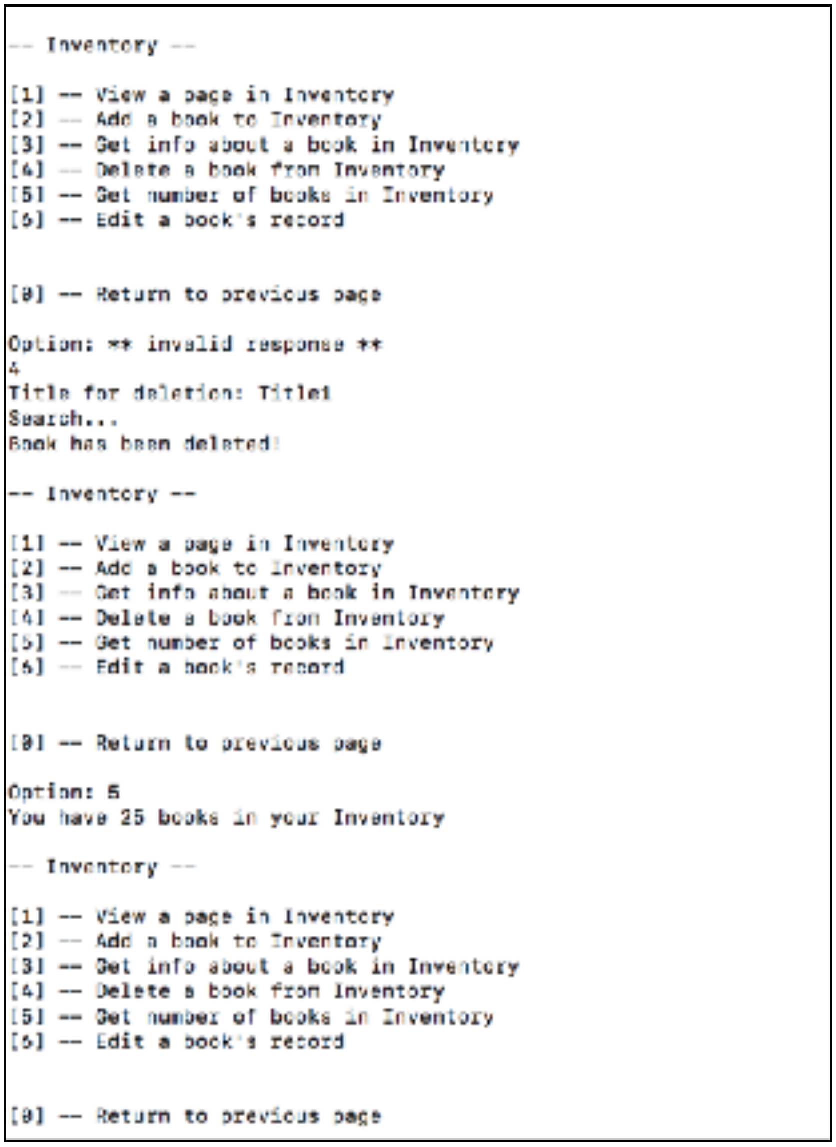
Pseudocode: The Inventory Module accepts a data file name, and with the correct data foaming it reads a series of books into an array of Book object pointers. These pointers can be manipulated, shifted, and deleted using member functions.
User can add book, which creates a new Book pointer, can edit the book, which accesses the dereferenced pointer an changes information, and delete books, which removes the pinter and downshifts the remaining array.
User can find a book's index in the inventory for the sake of calling its pointer directly.



```
Last login: Mon Jun 18 12:21:18 on ttys002
[illegible] ./Users/[illegible]/Library/Developer/Xcode/DerivedData/CIS[illegible]_FinalProject-[illegible]/Build/Products/Debug/CIS2
/Users/[illegible]/Desktop/datafile.txt
Choose from the following menu options:
1. Buy BOOKS
2. Manage inventory
3. View reports
4. Exit
Option: 2

      -- Inventory --

[1] -- View a page in Inventory
[2] -- Add a book to Inventory
[3] -- Get info about a book in Inventory
[4] -- Delete a book from inventory
[5] -- Get number of books in Inventory
[6] -- Edit a book's record


[0] -- Return to previous page

Option: 1
There are 3 in your Inventory.
View Page: 1
Page 1
=========================================================================
Title                       Author              Publisher        Stock
-------------------------------------------------------------------------
Title1                       Author1            Publisher1         1
Title2                       Author1            Publisher1         1
Title3                       Author1            Publisher1         1
Title4                       Author3            Publisher1         1
Title5                       Author1            Publisher1         4
Title6                       Author?            Publisher1         1
Title7                       Author?            Publisher1         1
Title8                       Author3            Publisher1         ?
Title9                       Author3            Publisher1         1
Title10                      Author1            Publisher1         1

-- Inventory --

[1] -- View a page in Inventory
[2] -- Add a book to Inventory
[3] -- Get info about a book in Inventory
[4] -- Delete a book from inventory
[5] -- Get number of books in Inventory
[6] -- Edit a book's record
```



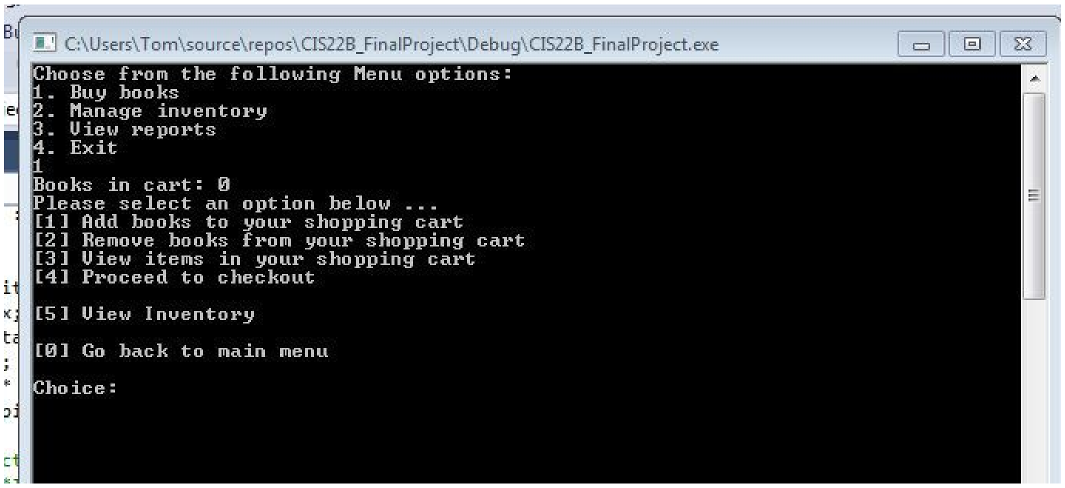
```
Option: 2
ISBN: 123456
Title: Tom and Jerry
Author: Jim Jake
Publisher: Penguin Pub.
Date Published (YYYYMMDD): 20180716
Stock: 2
Retail Price: $12.32
Wholesale Price: $11.68

-- Inventory --

[1] -- View a page in Inventory
[2] -- Add a book to Inventory
[3] -- Get info about a book in Inventory
[4] -- Delete a book from Inventory
[5] -- Get number of books in Inventory
[6] -- Edit a book's record


[0] -- Return to previous page

Option: 3
Title for info: Tom and Jerry
Search...
ISBN: 123456
Title: Tom and Jerry
Author: Jim Jake
Publisher: Penguin Pub.
Date Published: 20180716
Stock: 2
Retail Price: 12.3
Wholesale Price: 11.68
```



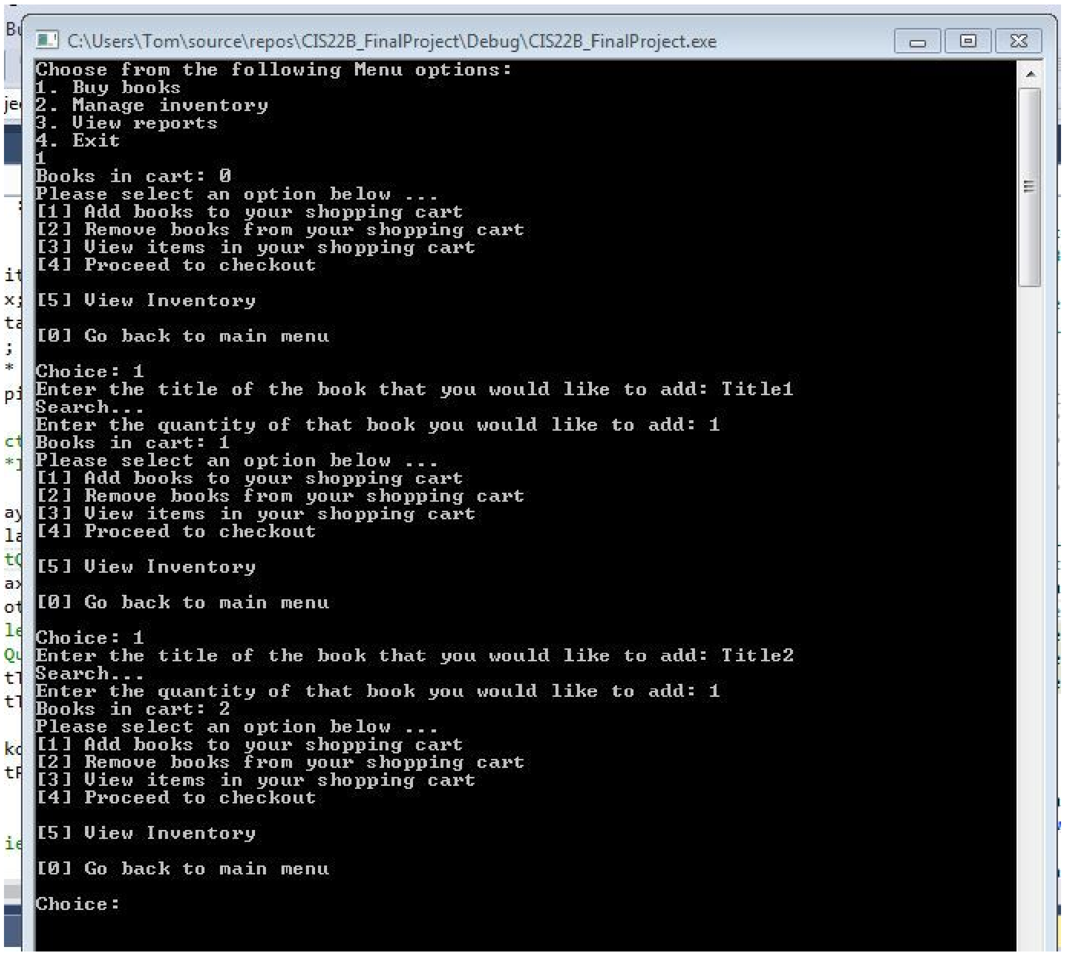
```
-- Inventory --

[1] -- View a page in Inventory
[2] -- Add a book to Inventory
[3] -- Get info about a book in Inventory
[4] -- Delete a book from Inventory
[5] -- Get number of books in Inventory
[6] -- Edit a book's record


[0] -- Return to previous page

Option: ** invalid response **
4
Title for deletion: Title1
Search...
Book has been deleted!

-- Inventory --

[1] -- View a page in Inventory
[2] -- Add a book to Inventory
[3] -- Get info about a book in Inventory
[4] -- Delete a book from Inventory
[5] -- Get number of books in Inventory
[6] -- Edit a book's record


[0] -- Return to previous page

Option: 5
You have 25 books in your Inventory

-- Inventory --

[1] -- View a page in Inventory
[2] -- Add a book to Inventory
[3] -- Get info about a book in Inventory
[4] -- Delete a book from Inventory
[5] -- Get number of books in Inventory
[6] -- Edit a book's record


[0] -- Return to previous page
```

Cashier Module
-int quantity // holds value of quantity -double tax // holds value of tax for total cost calculation -double total // holds total cost of all books in cart -Book book // Instance of book class object -Inventory *inventory // Pointer to inventory, to access inventory module -Cart shoppingCart // Instance of Cart class object
+Cashier(Inventory *): void // constructor +displayTopMenu(): int // function to accept user input +displayInternalMenu(int) : int // function to interpret output from displayTopMenu +setTax() : void // multiplies tax by total, adds tax to total to process transaction +setTotal() : void // adds cost of all books in cart +getTax() : double // return value of tax +getTotal() : double // return price of all books from cart, + tax +checkout() : void // function to process of books from cart, prints receipt +printReceipt() : void // prints copy of the transaction for user

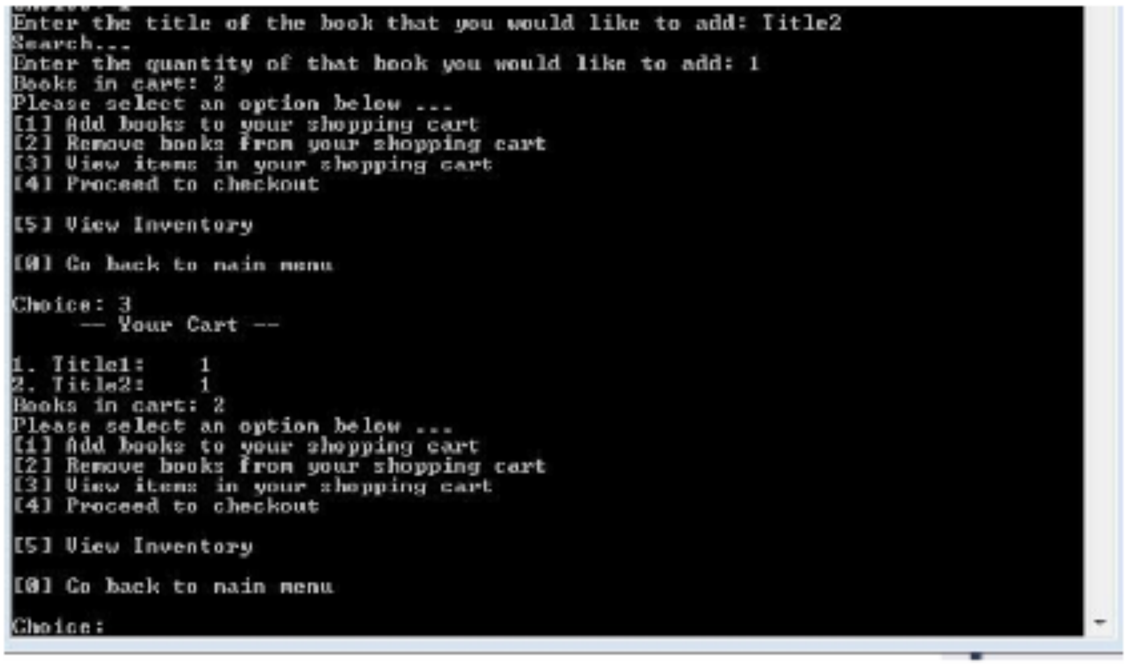
```
//The cashier module accepts user input from the main menu and
//prompts them to take actions towards purchasing books. The user
//selects an option from displayTopMenu, with options like adding
//or removing books, viewing their cart, and so on. Once the user
//provides input, that input is sent to displayInternalMenu where
//they can add books by title and the quantity of said book they wish
//to add/remove from their cart. When they select to checkout their
//cart, books (and the desired quantity) are removed from the inventory,
```



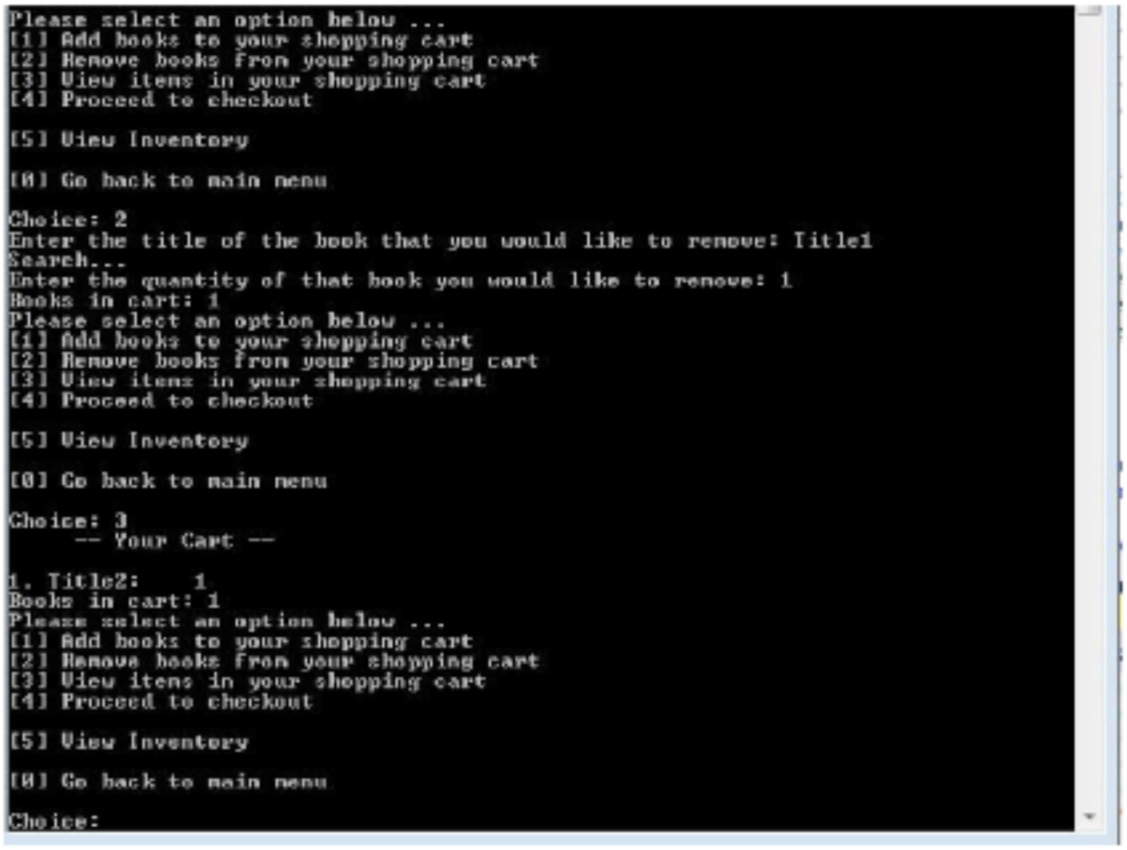
```
C:\Users\Tom\source\repos\CIS22B_FinalProject\Debug\CIS22B_FinalProject.exe
Choose from the following Menu options:
1. Buy books
2. Manage inventory
3. View reports
4. Exit
1
Books in cart: 0
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice:
```



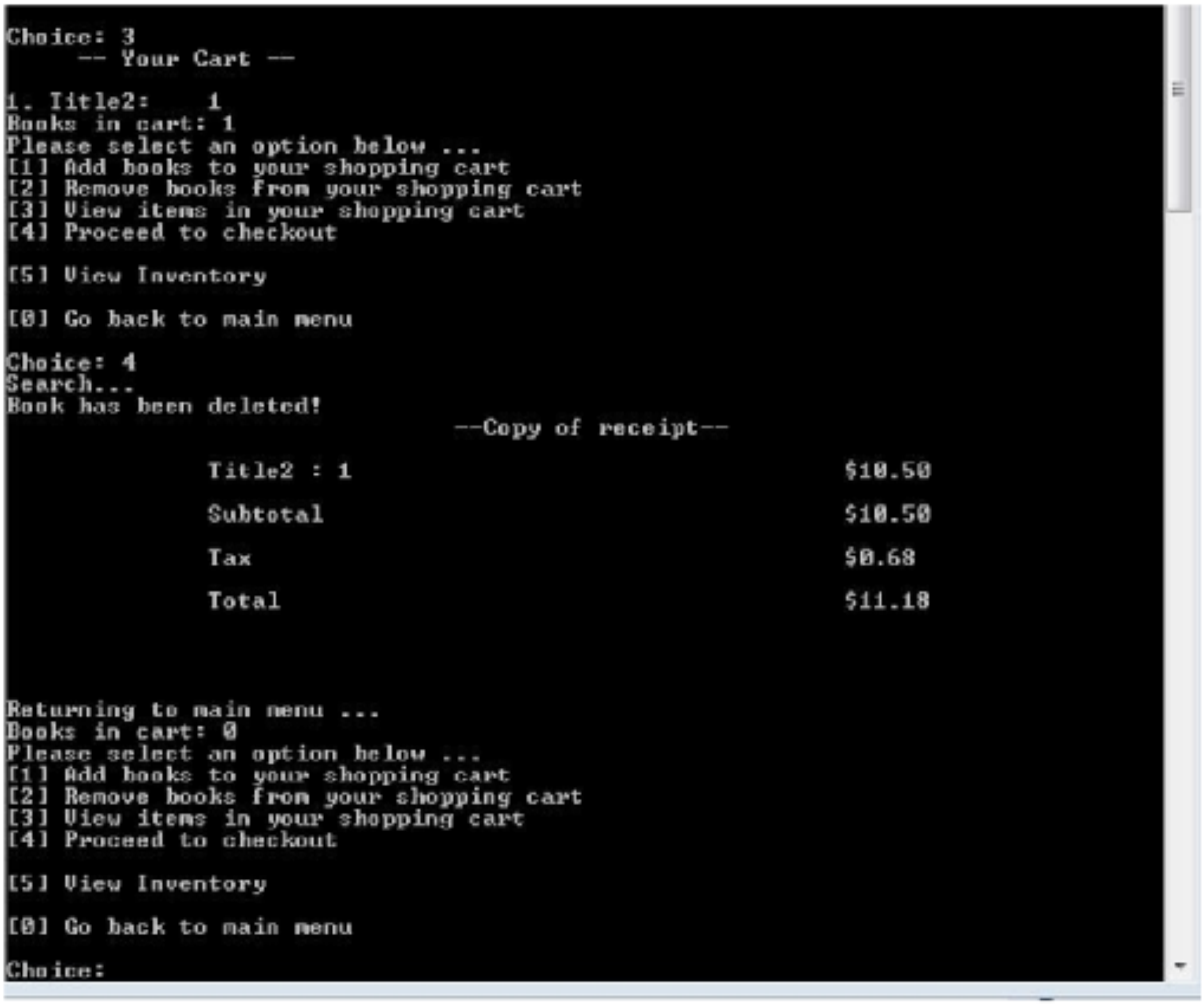
```
C:\Users\Tom\source\repos\CIS22B_FinalProject\Debug\CIS22B_FinalProject.exe
Choose from the following Menu options:
1. Buy books
2. Manage inventory
3. View reports
4. Exit
1
Books in cart: 0
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice: 1
Enter the title of the book that you would like to add: Title1
Search...
Enter the quantity of that book you would like to add: 1
Books in cart: 1
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice: 1
Enter the title of the book that you would like to add: Title2
Search...
Enter the quantity of that book you would like to add: 1
Books in cart: 2
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice:
```



```
Enter the title of the book that you would like to add: Title2
Search...
Enter the quantity of that book you would like to add: 1
Books in cart: 2
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice: 3
      -- Your Cart --

1. Title1:    1
2. Title2:    1
Books in cart: 2
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice:
```

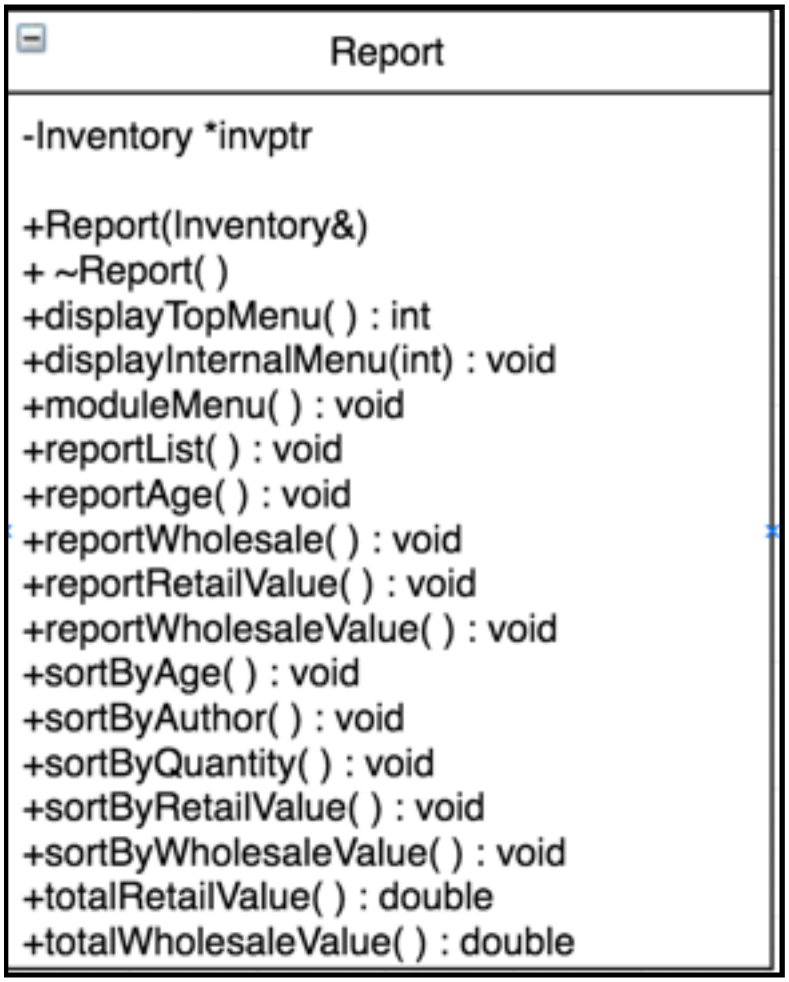


```
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice: 2
Enter the title of the book that you would like to remove: Title1
Search...
Enter the quantity of that book you would like to remove: 1
Books in cart: 1
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice: 3
      -- Your Cart --

1. Title2:    1
Books in cart: 1
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice:
```



```
Choice: 3
     -- Your Cart --

1. Title2:    1
Books in cart: 1
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice: 4
Search...
Book has been deleted!
                               --Copy of receipt--

              Title2 : 1                                    $10.50

              Subtotal                                      $10.50

              Tax                                           $0.68

              Total                                         $11.18



Returning to main menu ...
Books in cart: 0
Please select an option below ...
[1] Add books to your shopping cart
[2] Remove books from your shopping cart
[3] View items in your shopping cart
[4] Proceed to checkout

[5] View Inventory

[0] Go back to main menu

Choice:
```

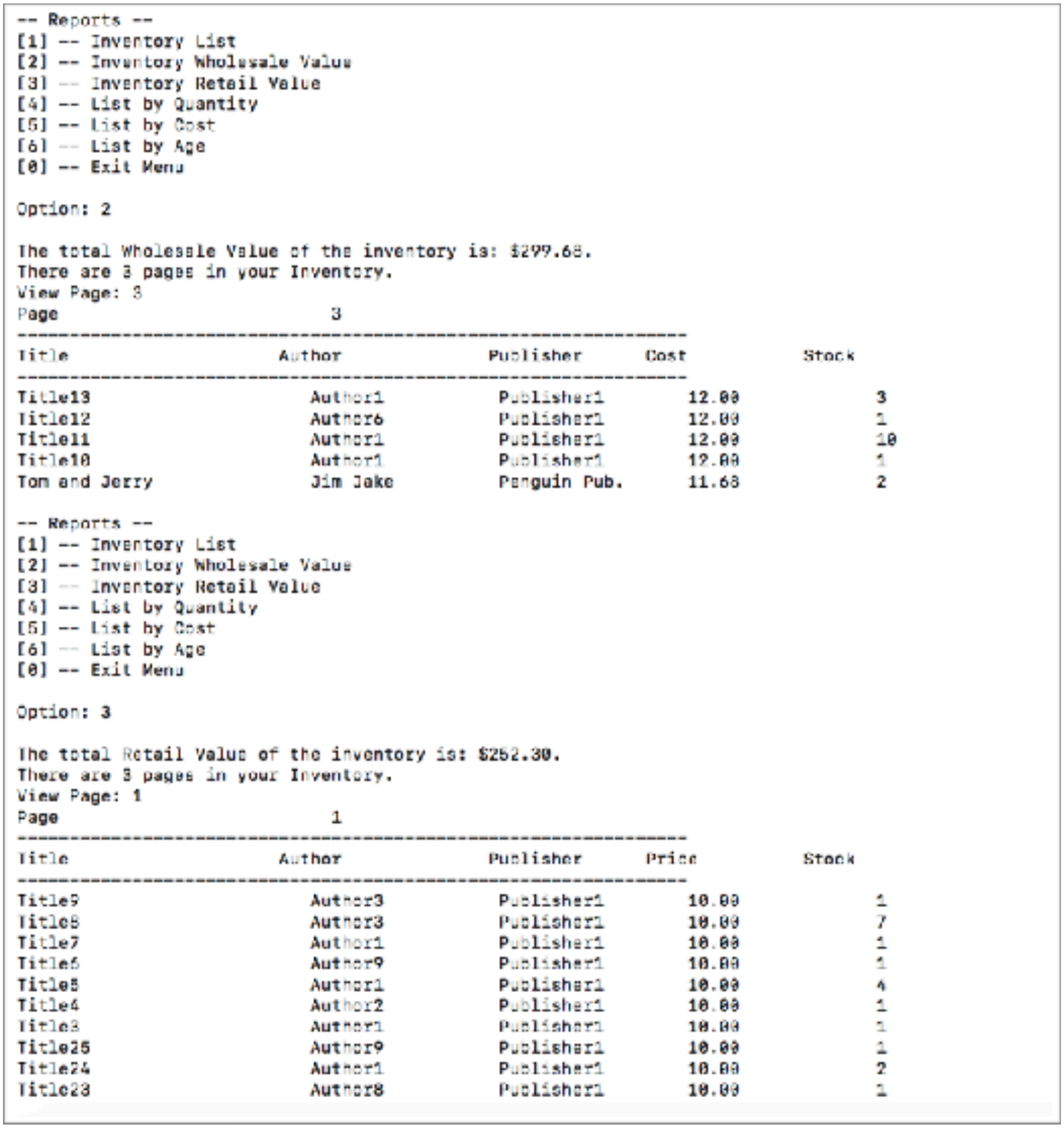
Report
UML and


Report

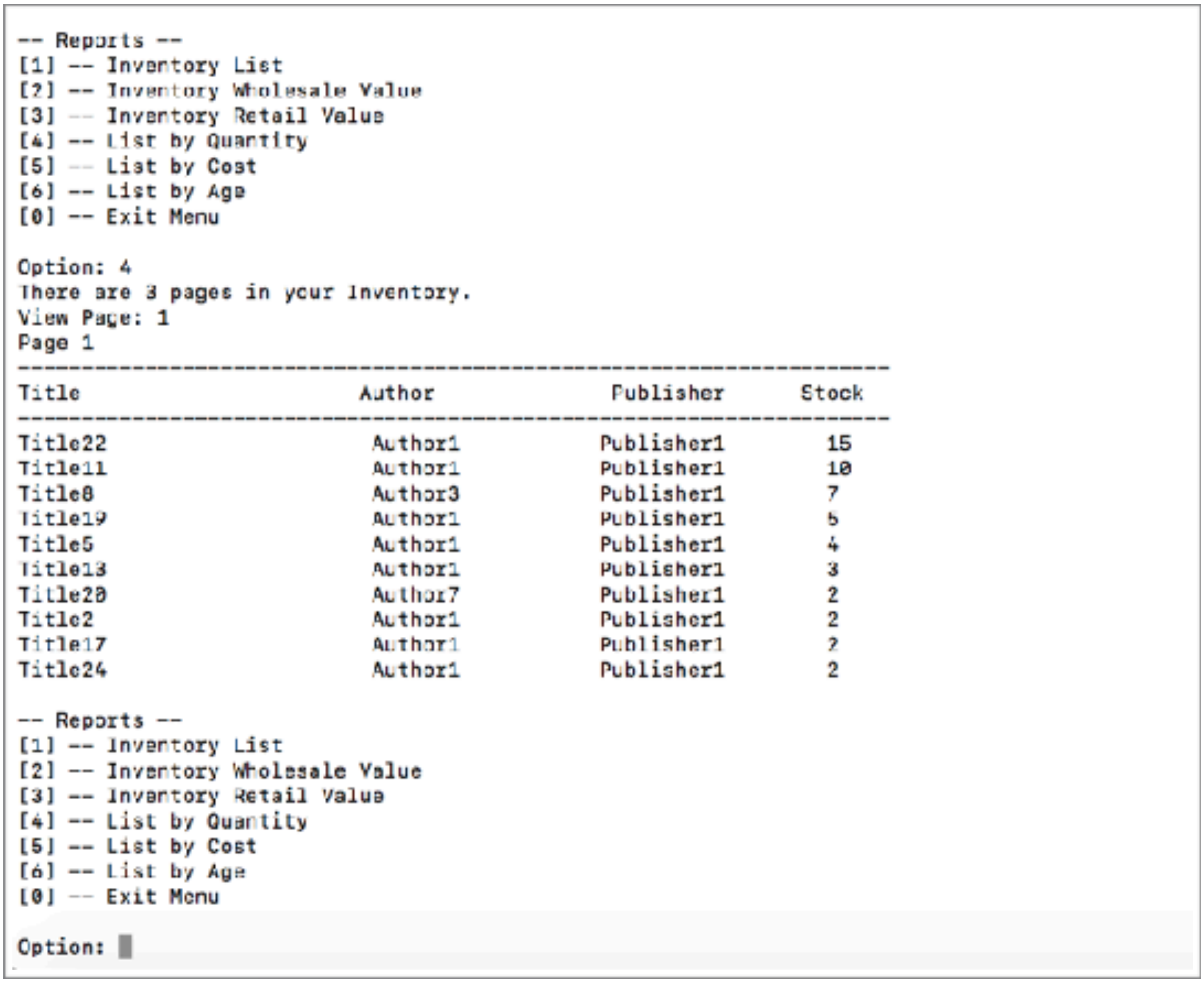
```
-Inventory *invptr

+Report(Inventory&)
+ ~Report( )
+displayTopMenu( ) : int
+displayInternalMenu(int) : void
+moduleMenu( ) : void
+reportList( ) : void
+reportAge( ) : void
+reportWholesale( ) : void
+reportRetailValue( ) : void
+reportWholesaleValue( ) : void
+sortByAge( ) : void
+sortByAuthor( ) : void
+sortByQuantity( ) : void
+sortByRetailValue( ) : void
+sortByWholesaleValue( ) : void
+totalRetailValue( ) : double
+totalWholesaleValue( ) : double
```

Screenshots

```
Class Report to display to user user specified total and or user
specified type of sort sorted inventory.
```

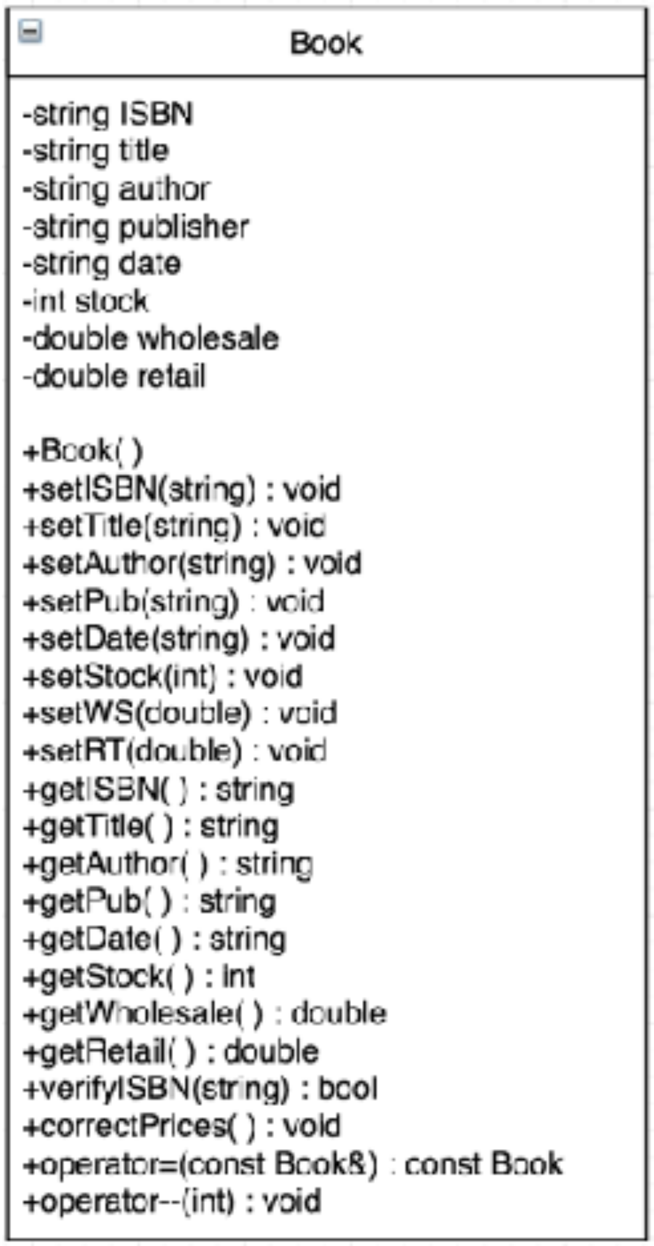


```
-- Reports --
[1] -- Inventory List
[2] -- Inventory Wholesale Value
[3] -- Inventory Retail Value
[4] -- List by Quantity
[5] -- List by Cost
[6] -- List by Age
[0] -- Exit Menu

Option: 2

The total Wholesale Value of the inventory is: $299.68.
There are 3 pages in your Inventory.
View Page: 3
Page                              3
----------------------------------------------------------------
Title                    Author              Publisher      Cost           Stock
----------------------------------------------------------------
Title13                     Author1             Publisher1         12.00            3
Title12                     Author6             Publisher1         12.00            1
Title11                     Author1             Publisher1         12.00            10
Title10                     Author1             Publisher1         12.00            1
Tom and Jerry               Jim Jake            Penguin Pub.       11.68            2

-- Reports --
[1] -- Inventory List
[2] -- Inventory Wholesale Value
[3] -- Inventory Retail Value
[4] -- List by Quantity
[5] -- List by Cost
[6] -- List by Age
[0] -- Exit Menu

Option: 3

The total Retail Value of the inventory is: $252.30.
There are 3 pages in your Inventory.
View Page: 1
Page                              1
----------------------------------------------------------------
Title                    Author              Publisher      Price          Stock
----------------------------------------------------------------
Title9                      Author3             Publisher1         10.00            1
Title8                      Author3             Publisher1         10.00            7
Title7                      Author1             Publisher1         10.00            1
Title6                      Author9             Publisher1         10.00            1
Title5                      Author1             Publisher1         10.00            4
Title4                      Author2             Publisher1         10.00            1
Title3                      Author1             Publisher1         10.00            1
Title25                     Author9             Publisher1         10.00            1
Title24                     Author1             Publisher1         10.00            2
Title23                     Author8             Publisher1         10.00            1
```



```
-- Reports --
[1] -- Inventory List
[2] -- Inventory Wholesale Value
[3] -- Inventory Retail Value
[4] -- List by Quantity
[5] -- List by Cost
[6] -- List by Age
[0] -- Exit Menu

Option: 4
There are 3 pages in your Inventory.
View Page: 1
Page 1
--------------------------------------------------------------------
Title                         Author              Publisher      Stock
--------------------------------------------------------------------
Title22                        Author1            Publisher1       15
Title11                        Author1            Publisher1       10
Title8                         Author3            Publisher1       7
Title19                        Author1            Publisher1       6
Title5                         Author1            Publisher1       4
Title13                        Author1            Publisher1       3
Title20                        Author7            Publisher1       2
Title2                         Author1            Publisher1       2
Title17                        Author1            Publisher1       2
Title24                        Author1            Publisher1       2

-- Reports --
[1] -- Inventory List
[2] -- Inventory Wholesale Value
[3] -- Inventory Retail Value
[4] -- List by Quantity
[5] -- List by Cost
[6] -- List by Age
[0] -- Exit Menu

Option: 
```

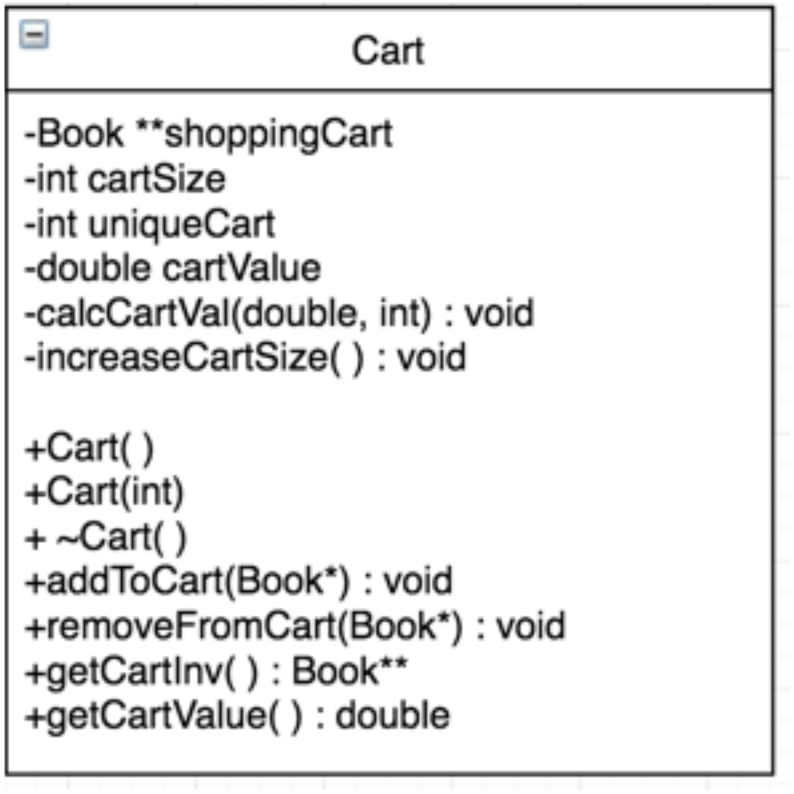
Additional Classes UML



Book

-string ISBN
-string title
-string author
-string publisher
-string date
-int stock
-double wholesale
-double retail

+Book( )
+setISBN(string) : void
+setTitle(string) : void
+setAuthor(string) : void
+setPub(string) : void
+setDate(string) : void
+setStock(int) : void
+setWS(double) : void
+setRT(double) : void
+getISBN( ) : string
+getTitle( ) : string
+getAuthor( ) : string
+getPub( ) : string
+getDate( ) : string
+getStock( ) : int
+getWholesale( ) : double
+getRetail( ) : double
+verifyISBN(string) : bool
+correctPrices( ) : void
+operator=(const Book&) : const Book
+operator--(int) : void

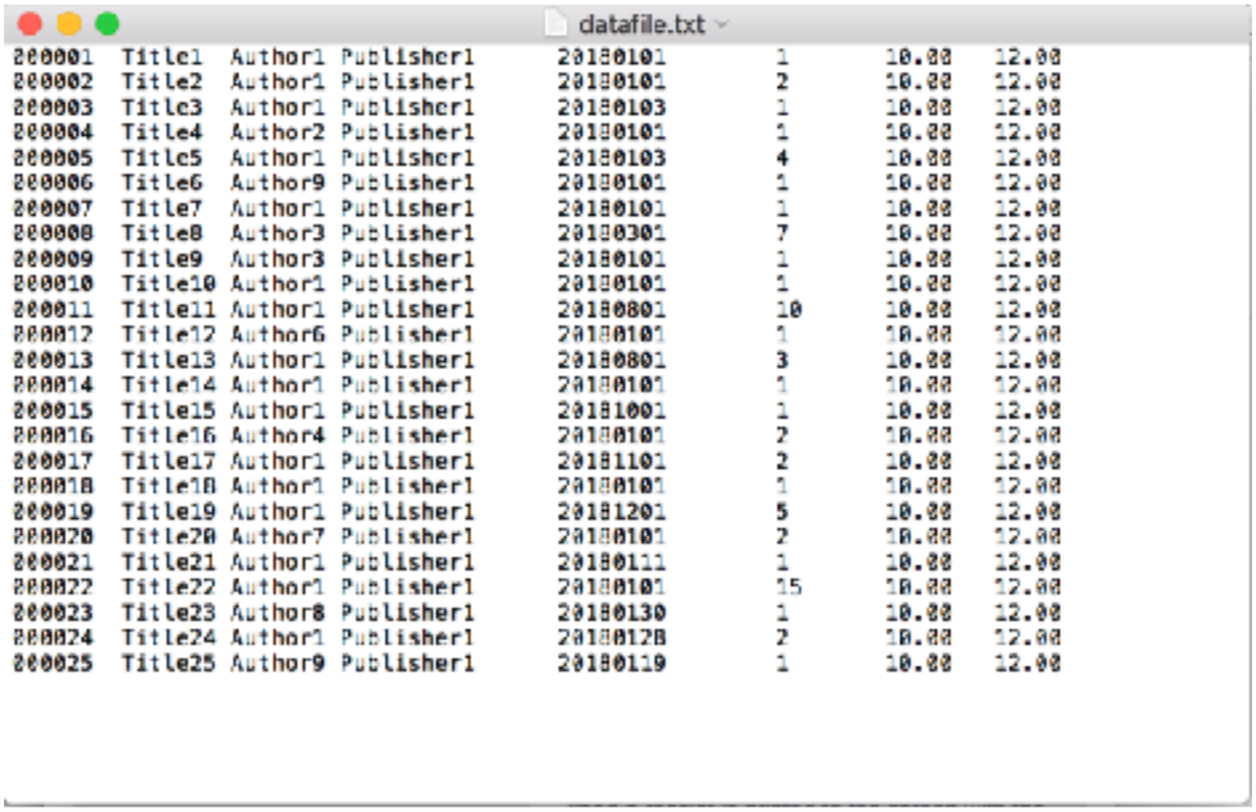


```
Cart

-Book **shoppingCart
-int cartSize
-int uniqueCart
-double cartValue
-calcCartVal(double, int) : void
-increaseCartSize( ) : void

+Cart( )
+Cart(int)
+ ~Cart( )
+addToCart(Book*) : void
+removeFromCart(Book*) : void
+getCartInv( ) : Book**
+getCartValue( ) : double
```

Module

+Module() // constructor
+~Module() // destructor
+virtual displayTopMenu() : int // displays top menu and returns user input for selection
+virtual displayInternalMenu(int) : void // displays internal menu based on an integer choice

Data File Used for Testing



```
000001   Title1   Author1 Publisher1       20180101        1        10.00    12.00
000002   Title2   Author1 Publisher1       20180101        2        10.00    12.00
000003   Title3   Author1 Publisher1       20180103        1        10.00    12.00
000004   Title4   Author2 Publisher1       20180101        1        10.00    12.00
000005   Title5   Author1 Publisher1       20180103        4        10.00    12.00
000006   Title6   Author9 Publisher1       20180101        1        10.00    12.00
000007   Title7   Author1 Publisher1       20180101        1        10.00    12.00
000008   Title8   Author3 Publisher1       20180301        7        10.00    12.00
000009   Title9   Author3 Publisher1       20180101        1        10.00    12.00
000010   Title10  Author1 Publisher1       20180101        1        10.00    12.00
000011   Title11  Author1 Publisher1       20180801        10       10.00    12.00
000012   Title12  Author6 Publisher1       20180101        1        10.00    12.00
000013   Title13  Author1 Publisher1       20180801        3        10.00    12.00
000014   Title14  Author1 Publisher1       20180101        1        10.00    12.00
000015   Title15  Author1 Publisher1       20181001        1        10.00    12.00
000016   Title16  Author4 Publisher1       20180101        2        10.00    12.00
000017   Title17  Author1 Publisher1       20181101        2        10.00    12.00
000018   Title18  Author1 Publisher1       20180101        1        10.00    12.00
000019   Title19  Author1 Publisher1       20181201        5        10.00    12.00
000020   Title20  Author7 Publisher1       20180101        2        10.00    12.00
000021   Title21  Author1 Publisher1       20180111        1        10.00    12.00
000022   Title22  Author1 Publisher1       20180101        15       10.00    12.00
000023   Title23  Author8 Publisher1       20180130        1        10.00    12.00
000024   Title24  Author1 Publisher1       20180128        2        10.00    12.00
000025   Title25  Author9 Publisher1       20180119        1        10.00    12.00
```