**Sarah Parker**

**CPSC5010**

**Spring 2025**

**Final Project**

**4/17/2025**

**Link to YouTube –**

The bookstore management system is software that allows users to effectively manage a catalog of books through several different options via a user interface. The following UML diagram shows the classes, their members and functions, and the relationships between them.

A diagram of a software system

AI-generated content may be incorrect.

**Entity Class:**

The entity class was designed as an abstract class with a pure virtual function called display. The classes book and transaction directly inherit from this class and implement the display method.

A computer screen with text

AI-generated content may be incorrect.

**Book Class:**

The Book class inherits from the Entity class is designed to contain all the information for a given book. Its private members include title, author, price, stock count, and a static member named bookCount. The public constructor functions include a parameterized constructor, a copy constructor, and a destructor. Public accessor methods exist to obtain the title, author, price, and stock count. The public mutators allow the price of a book to be set and the stock number to be set. The virtual function, display, is implemented. Finally, there is a static method called getBookCount, that will return the number of book objects that have been created. A bool value was placed as a parameter in the constructor to create this method. This way, when the copy constructor is called in the Node class, the bool value can be changed to false to prevent duplication of the bookCount value.

Header File

A screen shot of a computer program

AI-generated content may be incorrect.

CPP File

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.

**Transaction Class**

The Transaction class inherits from the Entity class and is used to help process book transactions. It has an association relationship with the Book class as it uses a pointer to the Book object in its constructor. It also has a composition relationship with the Date class as it uses the Date object as a pass-by-value parameter in its constructor. Since it inherits from Entity, an implementation for the virtual function, display, was implemented.

Header file:

A screen shot of a computer program

AI-generated content may be incorrect.

CPP File:

A screen shot of a computer screen

AI-generated content may be incorrect.

**Date Class**

Header file:

A screen shot of a computer program

AI-generated content may be incorrect.

CPP file:

A screen shot of a computer code

AI-generated content may be incorrect.

**Node Class:**

This class creates nodes for a doubly linked list, which ensures proper tracking of each object when books are added or removed from the catalog. The node class was created inside the Catalog file as it is a composition relationship, and the Catalog class requires its use. Its public members include a pointer to a book object, a pointer to the next node, and a pointer to the previous node. It creates a copy of the book by utilizing the book copy constructor.

**Catalog Class:**

This class utilizes the data structure of a doubly linked list to manage the books in the catalog. It accomplishes this by creating private members that are pointers to two nodes, one being the head (the beginning book of the catalog) and the other being the tail (the last book of the catalog). The public functions include a constructor and destructor. Additionally, there are public functions to add a book, search for a book, display the books, remove a book, decrease the stock count, and find the lowest-priced book.

The function to add a book does this by creating a new node at the end of the linked list. It then updates the pointer to ensure the previous tail node is now set to the “prev” pointer, and the newest node that was created is now the tail pointer.

The function to remove a book does this by searching for the book title among the nodes. If the book is the head node, then it will make the next node the new head node. If the book is the tail node then it will update the previous node as the tail node. Otherwise, if it is a middle node, it will update the previous node to point directly to the next node, therefore successfully eliminating the pointer to the node containing the book. Finally, the node is deleted to free memory.

Two other functions utilize the nodes in this class. The decreaseStock function is called when a transaction is processed. It will first find the node containing the book by the title, then get the stock number. Finally, it will update the stock number -1 to reduce the stock count for the book.

The function, findLowestPriceBook, will look through the nodes and initially assign the head node’s book price as the lowest price. It will then continue to traverse through the nodes and compare the price, if it finds a lower price it will assign that node as the lowestPrice. It will then return the node with the book with the lowest price.

Header File

A screen shot of a computer program

AI-generated content may be incorrect.

CPP File

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

The final file, Bookstore, in this project contains the main method to run the program. Initially, two books were created and added to the catalog. A third book was added via copy constructor, essentially creating a duplicate of the first book. A user interface was created by using a while loop that, while true, would allow the user to pick from the options until they requested to exit the program. The program options were created by using the case/switch function.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

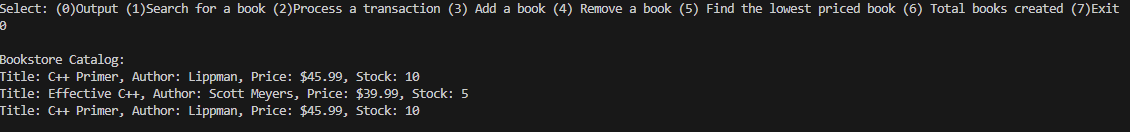
**Program Output**

Compilation and Execution

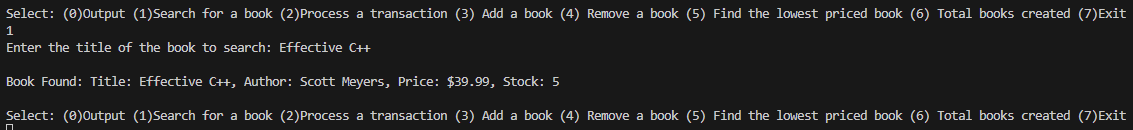
**A black screen with white text

AI-generated content may be incorrect.**

Selection 0 - Output

****

Selection 1 – Search for a book

****

Selection 2 – Process a Transaction – with stock count updated

**A black screen with white text

AI-generated content may be incorrect.**

Selection 3 – Add a book, followed by selection 0 (output to show updated catalog)

A screen shot of a computer

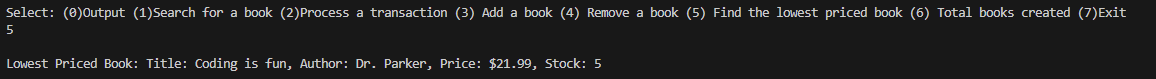
AI-generated content may be incorrect.

Selection 4 – Remove a book, followed by selection 0 (output to show updated catalog)

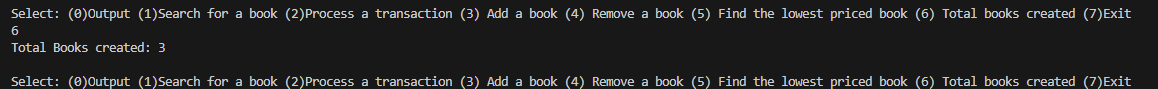
A black screen with white text

AI-generated content may be incorrect.

Selection 5 – Find the lowest priced book



Selection 6 – Total books created (discrete number of books in the catalog currently) - utilizing static method



Selection 7 – Exiting the program

