**OOP C++ Programming**

**Final Project Report**

**Bookshop Inventory System**



**Submitted By:**

**Jawad Manzoor (221567)**

**BS-IT -A**

**Submitted to:**

**Professor Dr. Abdul Hameed**

**Introduction:**

* The BookShop Inventory System is a simple console-based application developed using C++ language.
* The system allows users to manage the inventory of books in a bookshop.
* Users can add new books to the inventory, sell books, search for books based on a search term, view all books in the inventory, and save all books to a file.

**Features:**

The BookShop Inventory System has the following features:

1. **Add book**:

* Users can add new books to the inventory by entering the book ID, title, author, quantity, and price.
* The system can hold up to 100 books.

1. **Sell book**:
   * Users can sell books by entering the book ID and the quantity sold.
   * The system will update the quantity of the book in the inventory if there is enough stock, and output a success message.
   * Otherwise, an error message will be displayed.
2. **Search book**:
   * Users can search for books in the inventory based on a search term.
   * The search term can match the book ID, title, or author.
   * If a match is found, the book information will be displayed. Otherwise, an error message will be displayed.

4. **View all books**:

* Users can view all books in the inventory.
* The system will display the book ID, title, author, quantity, and price.

5. **Save books to file**:

* Users can save all books in the inventory to a file named "books.txt".

**Implementation:**

* The BookShop Inventory System is implemented using two classes: Book and BookShop. The Book class defines the properties of a book, such as book ID, title, author, quantity, and price.
* The BookShop class defines the inventory of books as an array of Book objects. It also provides methods for adding a new book, selling a book, searching for a book, viewing all books, and saving all books to a file.
* The main function of the program displays a menu to the user with options to add a book, sell a book, search for a book, view all books, save books to file, or exit the program.
* The user's choice is read in from the command line and the appropriate method of the BookShop object is called.
* When the user chooses to add a book, the program prompts the user to enter the book information and adds the book to the inventory.
* If the maximum number of books is reached, an error message is displayed.
* When the user chooses to sell a book, the program prompts the user to enter the book ID and the quantity sold.
* The program searches the inventory for the book with the given ID and updates its quantity if there is enough stock.
* Otherwise, an error message is displayed.
* When the user chooses to search for a book, the program prompts the user to enter a search term. The program searches the inventory for any books whose ID, title, or author matches the search term. Ifone or more books are found, the program displays their information. Otherwise, an error message is displayed.
* When the user chooses to view all books, the program displays the information of all books in the inventory.
* When the user chooses to save all books to a file, the program writes the book information in a file named "books.txt".

**Source Code:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252 | #include <iostream>  #include <fstream>  #include <string>  **using** **namespace** std;  *// Define a Book class*  **class** **Book** {  private:  string book\_id;  string book\_title;  string book\_author;  int book\_quantity;  double book\_price;  public:  *// Constructor for Book class*  Book(string id, string title, string author, int quantity, double price)  {  book\_id = id;  book\_title = title;  book\_author = author;  book\_quantity = quantity;  book\_price = price;  }  *// Getter methods for Book class*  string get\_id() {  **return** book\_id;  }  string get\_title() {  **return** book\_title;  }  string get\_author() {  **return** book\_author;  }  int get\_quantity() {  **return** book\_quantity;  }  double get\_price() {  **return** book\_price;  }  *// Setter method for book quantity*  void set\_quantity(int quantity) {  book\_quantity = quantity;  }  };  *// Define a BookShop class*  **class** **BookShop** {  private:  Book\* books[100]; *// Array of Book pointers to hold up to 100 books*  int num\_books; *// Number of books currently in the array*  public:  *// Constructor for BookShop class*  BookShop() {  num\_books = 0;  }  *// Method to add a new book to the// array*  void add\_book(string id, string title, string author, int quantity, double price)  {  **if** (num\_books < 100) {  Book\* new\_book = **new** Book(id, title, author, quantity, price);  *// Create a new Book object*    books[num\_books] = new\_book; *// Add the new Book object to the array*    num\_books++; *// Increment the number of books in the array*  } **else** {  cout << "Error: Maximum number of books reached" << endl;  }  }  *// Method to sell a book*  void sell\_book(string id, int quantity)  {  bool found = false;  **for** (int i = 0; i < num\_books; i++)  {  **if** (books[i]->get\_id() == id)  { *// If the book with the given ID is found*    int new\_quantity = books[i]->get\_quantity() - quantity;    *// Calculate the new quantity*  **if** (new\_quantity < 0)  {  *// If there is not enough stock, output an error message*    cout << "Error: Not enough stock" << endl;    } **else**  {  *// Otherwise, update the quantity and output a success message*    books[i]->set\_quantity(new\_quantity);  cout << "Sale successful" << endl;  }  found = true; *// Set found to true*  **break**; *// Exit the loop*  }  }  **if** (!found)  {    *// If the book is not found, output an error message*    cout << "Error: Book not found" << endl;  }  }  *// Method to search for a book based on a search term*  void search\_book(string search\_term)  {  bool found = false;  **for** (int i = 0; i < num\_books; i++)  {  Book\* book = books[i];  **if** (book->get\_id() == search\_term || book->get\_title() == search\_term || book->get\_author() == search\_term) { *// If the search term matches the ID, title or author of the book*  cout << "ID: " << book->get\_id() << endl;  cout << "Title: " << book->get\_title() << endl;  cout << "Author: " << book->get\_author() << endl;  cout << "Quantity: " << book->get\_quantity() << endl;  cout << "Price: $" <<book->get\_price() << endl << endl; *// Output the book information*  found = true; *// Set found to true*  }  }  **if** (!found)  {  *// If no books are found, output an error message*  cout << "No books found" << endl;  }  }  *// Method to view all books in the array*  void view\_all\_books() {  **for** (int i = 0; i < num\_books; i++) {  Book\* book = books[i];  cout << "ID: " << book->get\_id() << endl;  cout << "Title: " << book->get\_title() << endl;  cout << "Author: " << book->get\_author() << endl;  cout << "Quantity: " << book->get\_quantity() << endl;  cout << "Price: $" << book->get\_price() << endl << endl; *// Output the book information*  }  }  *// Method to save all books to a file*  void save\_books\_to\_file(string filename) {  ofstream output\_file(filename);  **for** (int i = 0; i < num\_books; i++) {  Book\* book = books[i];  output\_file << book->get\_id() << "," << book->get\_title() << "," << book->get\_author() << "," << book->get\_quantity() << "," << book->get\_price() << endl; *// Write the book information to the file*  }  output\_file.close(); *// Close the file*  }  };  *// Main function*  int main() {  BookShop bookshop; *// Create a new BookShop object*  int choice;  **do** {  *// Display the menu*  cout << "BookShop Inventory System" << endl;  cout << "1. Add book" << endl;  cout << "2. Sell book" << endl;  cout << "3. Search book" << endl;  cout << "4. View all books" << endl;  cout << "5. Save books to file" << endl;  cout << "6. Exit" << endl;  cout << "Enter your choice: ";  cin >> choice;  **switch** (choice) {  **case** 1: {  *// If the user selects option 1, prompt them to enter book information and add the book to the array*  string id, title, author;  int quantity;  double price;  cout << "Enter book ID: ";  cin >> id;  cout << "Enter book title: ";  cin.ignore();  getline(cin, title);  cout << "Enter book author: ";  getline(cin, author);  cout << "Enter book quantity: ";  cin >> quantity;  cout << "Enter book price: ";  cin >> price;  bookshop.add\_book(id, title, author, quantity, price);  cout << "Book added successfully" << endl;  **break**;  }  **case** 2:  {    *// If the user selects option 2, prompt them to enter book ID and quantity sold, and update the quantity of the book in the array*  string id;  int quantity;  cout << "Enter book ID: ";  cin >> id;  cout << "Enter quantity sold: ";  cin >> quantity;  bookshop.sell\_book(id, quantity);  **break** ;  }  **case** 3:  {  *// If the user selects option 3, prompt them to enter a search term and search for the book in the array*  string search\_term;  cout << "Enter search term: ";  cin.ignore();  getline(cin, search\_term);  bookshop.search\_book(search\_term);  **break**;  }  **case** 4:  {  *// If the user selects option 4, display all books in the array*  bookshop.view\_all\_books();  **break**;  }  **case** 5:  { *// If the user selects option 5, save all books to a file*  bookshop.save\_books\_to\_file("books.txt");  cout << "Books saved to file" << endl ;  }  **case** 6:  { *// If the user selects option 6, exit the program*  **break**;  }  default:  { *// If the user selects an invalid option, display an error message*  cout << "Invalid choice" << endl;  **break**;  }  }  cout << endl;  } **while** (choice != 6);  **return** 0;  } |

**Conclusion:**

The BookShop Inventory System is a simple but useful application for managing the inventory of books in a bookshop.

It provides basic features for adding, selling, searching, and viewing books.